THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

L'algorithme de biorthogonalisation de Lanczos, la méthode QMR et leur application à l'équation de Lyapunov

Denis, Christelle

Award date:
1999

Facultés Universitaires Notre-Dame de la Paix
Namur
Faculté des Sciences - Département de Mathématique

L'algorithme de biorthogonalisation de Lanczos, la méthode QMR et leur application à l'équation matricielle de Lyapunov.

Mémoire présenté pour l'obtention du grade
de Licenciée en Sciences
mathématiques
par

Promoteur : Mme S. Thiry.

Christelle DENIS

Année académique : 1998-1999

Je tiens à remercier tout particulièrement Madame Suzanne Thiry pour son aide, sa disponibilité et sa gentillesse.

Merci également à tous ceux et celles qui m'ont apporté leur soutien et leur présence tout au long de ma licence en mathématiques.

L'algorithme de biorthogonalisation de Lanczos, la méthode QMR et leur application à l'équation de Lyapunov

Résumé

L'algorithme de biorthogonalisation de Lanczos est un processus permettant d'engendrer deux séquences finies de vecteurs (vecteurs de Lanczos) reliés par une condition de biorthogonalité.

L'algorithme QMR pour la résolution des systèmes linéaires est une méthode itérative dont les itérés successifs sont exprimés dans une base constituée des vecteurs de Lanczos. Chaque pas de l'algorithme conduit à un problème de minimisation au sens des moindres carrés. Nous analyserons ici deux approches différentes pour l'implémentation de cet algorithme.

Finalement, nous présentons l'algorithme QMR pour la résolution de l'équation matricielle de Lyapunov $AX + XA^T = -DD^T$.

The Lanczos biorthogonalization algorithm, the QMR algorithm and their application to the Lyapunov equation

Abstract

The Lanczos biorthogonalization algorithm is a process generating two finite sequences of vectors (Lanczos vectors) satisfying a biorthogonality condition.

The QMR algorithm is an iterative method for solving linear systems where the iterates are linear combinations of the Lanczos vectors. Each step of the algorithm leads to a least square minimization problem. In this work, we present two different approaches for the QMR algorithm.

Finally we apply this algorithm to the Lyapunov matrix equation $AX + XA^T = -DD^T$.

Table des matières

Introduction

Le but de ce mémoire est d'analyser l'algorithme QMR. Nous verrons deux implementations différentes pour les systèmes linéaires et une application à l'équation de Lyapunov.

Le premier chapitre sera consacré à l'algorithme de biorthogonalisation de Lanczos qui engendre deux séquences finies de vecteurs de Lanczos obtenus par une récurrence à trois termes.

Nous développerons ensuite l'algorithme QMR ("Quasi Minimal Residual algorithm") qui permet de résoudre un système linéaire $Ax = b$.
Nous utiliserons les vecteurs de Lanczos construits par l'algorithme de biorthogonalisation. A chaque pas de l'algorithme, nous aboutirons à un problème de minimisation au sens des moindres carrés, résolu par une factorisation QR. Pour ces deux premiers chapitres, nous nous baserons principalement sur [7].

Une autre façon d'implémeter l'algortihme QMR est d'utiliser les vecteurs engendrés par l'algorithme de Lanczos basé sur un couple de récurrences à deux termes [5], et de résoudre le problème de minimisation au sens des moindres carrés par une récurrence simple plutôt que par une factorisation QR.
Cette variante de l'algorithme QMR, proposée récemment par M. Sauren et H.M. Bucker dans [12], sera présentée au chapitre 3.

Finalement le chapitre 4 sera consacré à une application de l'algorithme QMR à l'équation matricielle de Lyapunov [9].

Chapitre 1

L'algorithme de Lanczos

Nous allons décrire le processus de Lanczos de base non symétrique. Ce processus permet de construire des vecteurs de Lanczos à l'aide d'une récurrence à trois termes. Les coefficients de ces récurrences sont les élements d'une matrice tridiagonale.

1.1 Vecteurs de Lanczos

1.1.1 Définition

Soit $A \in \mathbb{C}^{N \times N}$ une matrice réelle ou complexe quelconque et B une matrice non singulière qui commute avec A.
La matrice B sera utilisée pour définir le "produit scalaire formel" :

$$< w, v >_B = w^* B v \qquad sur\ \mathbb{C}^{N \times N} \tag{1.1}$$

Nous l'appelons "produit scalaire formel", car si B n'est pas hermitienne définie positive, nous n'avons pas nécessairement un produit scalaire au sens classique du terme. Nous ne pouvons en effet garantir que :

$$\begin{aligned} \forall v, w \in \mathbb{C}^N \quad & < v, w > = < \overline{w, v} > \\ & < v, \cancel{w}\, v > \geq 0 \quad \text{si } v \neq 0. \end{aligned}$$

Nous utiliserons par la suite l'orthogonalité définie par

$$\forall v, w \in \mathbb{C}^N : v \perp_B w \Leftrightarrow < v, w >_B = 0$$

Un choix fréquent pour B est $B = I$ (la matrice identité), le produit scalaire devenant alors le produit scalaire euclidien classique sur $\mathbb{C}^{N \times N}$.

Y aurait-il intérêt à utiliser une autre matrice B

Choix de v_0 w_0.

L'identité :

$$< w, Av >_B = < A^* w, v >_B \tag{1.2}$$

résulte de la commutativité des matrices A et B.

L'algorithme de biorthogonalisation de Lanczos est un processus qui engendre, à partir d'une paire de vecteurs non orthogonaux v_0 et w_0, deux suites finies $\{v_n\}_{n=0}^{\nu-1}$ et $\{w_n\}_{n=0}^{\nu-1}$ de vecteurs de $\mathbb{C}^N$ caractérisées par :

- $v_n \in K_{n+1}(A, v_0) = span\{v_0, Av_0, ..., A^n v_0\}$
  $$w_m \in K_{m+1}(A^*, w_0) = span\{w_0, A^* w_0, ..., A^{*^m} w_0\} \tag{1.3}$$
- $$< w_m, v_n >_B = \begin{cases} 0 & \text{si } m \neq n \\ \delta_n & \text{si } m = n \end{cases} \tag{1.4}$$

Les vecteurs v_n sont appelés **vecteurs de Lanczos à droite**, et les vecteurs w_n, **vecteurs de Lanczos à gauche.**
Les sous-espaces $K_{n+1}(A, v_0)$ et $K_{n+1}(A^*, w_0)$ sont appelés **sous-espaces de Krylov.**

La longueur ν des suites dépend de A, B , v_0 et w_0. Elle résulte de l'impossibilité de prolonger les suites de sorte que (1.3) et (1.4) soient vérifiées avec $\delta_n \neq 0$. Ainsi nous avons $\delta_0, \delta_1, \cdots, \delta_{\nu-1} \neq 0$ et $\delta_\nu = 0$.

1.1.2 Propriété

Nous noterons, jusqu'à la fin de ce chapitre :

$$K_n(A, v_0) = K_n$$
$$K_n(A^*, w_0) = \tilde{K}_n$$

Proposition 1.1.1
Les sous-espaces K_n et $\tilde{K}_n$ ainsi que les vecteurs v_n et w_n définis par (1.3),(1.4) satisfont à :

1.

$$K_n \subseteq K_{n+1}, \quad n = 1, 2, \cdots$$
$$\tilde{K}_n \subseteq \tilde{K}_{n+1}, \quad n = 1, 2, \cdots \tag{1.5}$$

2.

$$K_{n+1} = span\{v_0, v_1, \cdots, v_n\}, \qquad n = 0, 1, \cdots$$
$$\tilde{K}_{n+1} = span\{w_0, w_1, ..., w_n\}, \qquad n = 0, 1, \cdots \tag{1.6}$$

3.

$$\forall w \in \tilde{K}_n : \ < w, v_n >_B = 0, \qquad 0 < n < \nu$$
$$\forall v \in K_n : \ < w_n, v >_B = 0, \qquad 0 < n < \nu \tag{1.7}$$

4.

$$v_n \in K_{n+1} \backslash K_n, \qquad 0 < n < \nu$$
$$w_n \in \tilde{K}_{n+1} \backslash \tilde{K}_n, \qquad 0 < n < \nu \tag{1.8}$$

Preuve :

1. La propriété d'emboîtement des sous-espaces K_n et $\tilde{K}_n$ résulte immédiatement de leur définition.

2. La preuve se fait par récurrence
   • Le cas $n = 0$ est évident.
   • Supposons que $n > 0$, $K_n = span\{v_0, v_1, ..., v_{n-1}\}$,
   et montrons que $K_{n+1} = span\{v_0, v_1, ..., v_n\}$
   Le vecteur $v_n \in K_{n+1}$ s'écrit

$$v_n = \sum_{i=0}^{n-1} \alpha_i v_i + \alpha_n A^n v_0 \tag{1.9}$$

   où $\alpha_n \neq 0$, sinon w_n est orthogonal à v_n et nous ne remplissons plus la condition (1.4).
   Nous pouvons écrire

$$A^n v_0 = \alpha_n^{-1}(v_n - \sum_{i=0}^{n-1} \alpha_i v_i)$$

   de sorte que $K_{n+1} = span\{v_0, \cdots, A^n v_0\} = span\{v_0, \cdots, v_{n-1}, v_n\}$.

3. C'est une conséquence de la condition de biorthogonalité (1.4) en prenant les sous-espaces de Krylov décrit en 2.

4. La propriété 4. est due au fait que dans la preuve de 2., nous avons $\alpha_n \neq 0$.

□

1.2 Récurrences pour les vecteurs de Lanczos

Nous allons utiliser les résultats précédents pour montrer que les vecteurs v_n et w_n satisfont chacun à une récurrence à trois termes.

Théorème 1.2.1
Pour $n = 0, \cdots, \nu - 1$:

$$\begin{aligned} Av_n &= v_{n+1}\gamma_n + v_n\alpha_n + v_{n-1}\beta_{n-1} \\ A^*w_n &= w_{n+1}\tilde{\gamma}_n + w_n\tilde{\alpha}_n + w_{n-1}\tilde{\beta}_{n-1} \end{aligned} \tag{1.10}$$

où nous avons posé $v_{-1} = w_{-1} = 0$ *et* $\beta_{-1} = \tilde{\beta}_{-1} = 0$

__Preuve__ :
Nous définissons les matrices d'Hessenberg d'ordre n T_n et $\tilde{T}_n$ de la façon suivante :

$$T_n := (\tau_{k,j})_{k,j=0}^{n-1} \quad et \quad \tilde{T}_n := (\tilde{\tau}_{k,j})_{k,j=0}^{n-1}.$$

Nous pouvons ainsi écrire grâce à (1.6) et (1.8)

$$Av_{n-1} = v_n\tau_{n,n-1} + v_{n-1}\tau_{n-1,n-1} + \cdots + v_0\tau_{0,n-1}$$

$$A^*w_{n-1} = w_n\tau_{n,n-1} + w_{n-1}\tau_{n-1,n-1} + \cdots + w_0\tau_{0,n-1} \tag{1.11}$$

où $\tau_{n,n-1} \neq 0, \tilde{\tau}_{n,n-1} \neq 0$.

En définissant pour $n \leq \nu$, $V_n := [v_0 \cdots v_{n-1}]$ et $W_n := [w_0 \cdots w_{n-1}]$, nous pouvons réécrire les relations (1.11) sous la forme suivante :

$$AV_\nu = V_\nu T_\nu + v_\nu\gamma_{\nu-1}e_\nu^T \tag{1.12}$$

$$A^*W_\nu = W_\nu\tilde{T}_\nu + w_\nu\tilde{\gamma}_{\nu-1}e_\nu^T \tag{1.13}$$

où e_n est le $n^{ème}$ vecteur unité.
Si nous notons $D_{\delta;n} := diag(\delta_0, \delta_1, \cdots, \delta_{n-1})$, nous pouvons réécrire la condition de biorthogonalité (1.4) sous la forme :

$$W_\nu^* BV_\nu = D_{\delta;\nu} \tag{1.14}$$

Par (1.14) et $v_\nu \perp_B \tilde{K}_\nu, w_\nu \perp_B K_\nu$, nous avons que

$$W_\nu^* B A V_\nu = D_{\delta;\nu} T_\nu, \qquad V_\nu^* B^* A^* W_\nu = \overline{D_{\delta;\nu}} \tilde{T}_\nu \tag{1.15}$$

ainsi

$$D_{\delta;\nu} T_\nu = \tilde{T}_\nu^* D_{\delta;\nu} \tag{1.16}$$

et donc T_ν et $\tilde{T}_\nu$ doivent être tridiagonales.

En posant
$\alpha_n = \tau_{n,n}$ pour $n = 0, \cdots, \nu - 1$
$\beta_n = \tau_{n,(n+1)}$ pour $n = 0, \cdots, \nu - 2$
$\gamma_n = \tau_{(n+1),n} \neq 0$ pour $n = 0, \cdots, \nu - 2$,

nous pouvons écrire

$$T_\nu = \begin{pmatrix} \alpha_0 & \beta_0 & 0 & \cdots & 0 \\ \gamma_0 & \alpha_1 & \beta_1 & \ddots & 0 \\ 0 & \gamma_1 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \ddots & \beta_{\nu-2} \\ 0 & 0 & \cdots & \gamma_{\nu-2} & \alpha_{\nu-1} \end{pmatrix} \tag{1.17}$$

et nous obtenons la relation (1.10).

□

Corollaire 1.2.1
Nous pouvons réécrire (1.10) sous forme matricielle :

$$AV_\nu = V_{\nu+1}\underline{T_\nu} \quad , \quad A^*W_\nu = W_{\nu+1}\underline{\tilde{T}_\nu} \tag{1.18}$$

en notant :

- $V_\nu = [v_0 v_1 \cdots v_{\nu-1}]\,,\ W_\nu = [w_0 w_1 \cdots w_{\nu-1}]$ (1.19)
  c'est-à-dire que les colonnes de la matrice V_ν sont constituées des vecteurs v_n.

- $$\underline{T_\nu} = \begin{pmatrix} T_\nu \\ \hline \\ \gamma_{\nu-1}e_\nu^T \end{pmatrix} = \begin{pmatrix} \alpha_0 & \beta_0 & 0 & \cdots & 0 \\ \gamma_0 & \alpha_1 & \beta_1 & \ddots & 0 \\ 0 & \gamma_1 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & \gamma_{\nu-2} & \alpha_{\nu-1} \\ 0 & 0 & 0 & 0 & \gamma_{\nu-1} \end{pmatrix} \tag{1.20}$$

- $\underline{\tilde{T}_\nu}$ a la même expression que $\underline{T_\nu}$ où les $\alpha_i, \beta_i, \gamma_i$ sont remplacés par $\tilde{\alpha}_i, \tilde{\beta}_i, \tilde{\gamma}_i$.

1.3 Détermination des paramètres des récurrences

Proposition 1.3.1

$$\alpha_n = \frac{< w_n, Av_n >_B}{\delta_n} \tag{1.21}$$

$$\alpha_n = \overline{\tilde{\alpha}_n} \tag{1.22}$$

$$\beta_{n-1} = \frac{\overline{\tilde{\gamma}_{n-1}}\delta_n}{\delta_{n-1}} \tag{1.23}$$

$$\tilde{\beta}_{n-1} = \frac{\overline{\gamma_{n-1}\delta_n}}{\delta_{n-1}} \tag{1.24}$$

<u>**Preuve**</u> :

- Nous trouvons α_n et β_{n-1} en effectuant les produits scalaires $< w_n, Av_n >$ et $< w_{n-1}, Av_n >$, et en nous servant de la proposition 1.1.1 et des relations (1.10).

$$\begin{aligned} \beta_{n-1} &= \frac{< w_{n-1}, Av_n >_B}{\delta_{n-1}} \\ &= \frac{< A^*w_{n-1}, v_n >_B}{\delta_{n-1}} \text{ par (1.2)} \\ &= \frac{\overline{\tilde{\gamma}_{n-1}}\delta_n}{\delta_{n-1}} \end{aligned} \tag{1.25}$$

- $\tilde{\alpha}_n$ et $\tilde{\beta}_{n-1}$ sont donnés par la relation (1.16).

  Elle s'écrit :

$$\begin{pmatrix} \delta_0\alpha_0 & \delta_0\beta_0 & 0 & \cdots \\ \delta_1\gamma_0 & \delta_1\alpha_1 & \delta_1\beta_1 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \delta_{\nu-2}\beta_{\nu-2} \\ 0 & \cdots & \delta_{\nu-1}\gamma_{\nu-2} & \delta_{\nu-1}\alpha_{\nu-1} \end{pmatrix} = \begin{pmatrix} \delta_0\overline{\tilde{\alpha}}_0 & \delta_1\overline{\tilde{\gamma}}_0 & 0 & \cdots \\ \delta_0\overline{\tilde{\beta}}_0 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \delta_{\nu-1}\overline{\tilde{\gamma}}_{\nu-2} \\ 0 & \cdots & \delta_{\nu-2}\overline{\tilde{\beta}}_{\nu-2} & \delta_{\nu-1}\overline{\tilde{\alpha}}_{\nu-1} \end{pmatrix}$$

Il faut donc que

$$\begin{aligned} \alpha_n &= \overline{\tilde{\alpha}_n} \quad \textit{pour } n = 0, \cdots, \nu-1 && (1.26) \\ \delta_n\beta_n &= \delta_{n+1}\overline{\tilde{\gamma}_n} \quad \textit{pour } n = 0, \cdots, \nu-2 && (1.27) \\ \delta_n\gamma_{n-1} &= \delta_{n-1}\overline{\tilde{\beta}_{n-1}} \quad \textit{pour } n = 0, \cdots, \nu-2 && (1.28) \end{aligned}$$

□

En éliminant δ_{n+1} dans les relations (1.28) et (1.27), nous obtenons :

$$\tilde{\beta}_n\tilde{\gamma}_n = \overline{\beta_n\gamma_n} \quad \textit{pour } n = 0, \cdots, \nu-2. \tag{1.29}$$

Un façon de satisfaire (1.29) serait de choisir :

$$\tilde{\beta}_n := \overline{\beta_n} \ , \ \tilde{\gamma}_n := \overline{\gamma_n} \quad , \text{ ce choix est équivalent à } \ \tilde{T}_\nu := \overline{T}_\nu. \tag{1.30}$$

Un autre choix possible est donné par :

$$\tilde{\beta}_n := \overline{\gamma_n} \ , \ \tilde{\gamma}_n := \overline{\beta_n} \quad , \text{ ce choix est équivalent à } \ \tilde{T}_\nu := T_\nu^*. \tag{1.31}$$

1.4 L'algorithme de biorthogonalisation de Lanczos

1.4.1 Algorithme général

- Initialisation

Choisir v_0, w_0 tels que $\delta_0 =< w_0, v_0 >_B \neq 0$ (1)

$\beta_{-1} = 0$ (2)

$n = 0$

- Itération $n+1$

1. Calculs

$\alpha_n = \frac{<w_n, Av_n>_B}{\delta_n}$; $\tilde{\alpha_n} = \overline{\alpha_n}$, (3)

IF $n > 0$ THEN

$\beta_{n-1} = \overline{\tilde{\gamma}_{n-1}}\delta_n/\delta_{n-1}$

$\tilde{\beta}_{n-1} = \overline{\gamma_{n-1}\delta_n/\delta_{n-1}} = \overline{\beta_{n-1}\gamma_{n-1}}/\tilde{\gamma}_{n-1}$, (4)

ENDIF

$v_{temp} = Av_n - v_n\alpha_n - v_{n-1}\beta_{n-1}$

$w_{temp} = A^* w_n - w_n\tilde{\alpha}_n - w_{n-1}\tilde{\beta}_{n-1}$ (5)

$\delta_{temp} =< w_{temp}, v_{temp} >_B$ (6)

2. Test

IF $\delta_{temp} = 0$ $THEN$ (7)

choisir $\gamma_n \neq 0, \tilde{\gamma}_n \neq 0$

$\nu = n+1$

$v_\nu = \frac{v_{temp}}{\gamma_n}$

$w_\nu = \frac{w_{temp}}{\tilde{\gamma}_n}$ (8)

$\delta_{n+1} = 0$ (9)

STOP

ELSE

choisir $\gamma_n \neq 0, \tilde{\gamma}_n \neq 0$

δ_{n+1} tel que $\gamma_n\overline{\tilde{\gamma}_n}\delta_{n+1} = \delta_{temp}$

$v_{n+1} = \frac{v_{temp}}{\gamma_n}$ (10)

$w_{n+1} = \frac{w_{temp}}{\tilde{\gamma}_n}$

n=n+1 (11)

Théorème 1.4.1
Les deux suites $\{v_n\}_{n=0}^{\nu-1}$ et $\{w_n\}_{n=0}^{\nu-1}$ générées par l'algorithme de biorthogonalisation satisfont à (1.1) et (1.2).
De plus, pour $m = \nu$ ou $n = \nu$, (1.3) et (1.4) sont satisfaits mais $\delta_\nu = 0$.

Les conditions (1.3) et (1.4) déterminent $\{v_n\}_{n=0}^{\nu-1}$ et $\{w_n\}_{n=0}^{\nu-1}$ à un facteur multiplicatif près.

Preuve :

La preuve se fait par récurrence.
Nous avons que $\{v_n\}_{n=0}^{\nu-1}$ et $\{w_n\}_{n=0}^{\nu-1}$ générées par l'algorithme de Lanczos satisfont à ces deux conditions pour $n = 0, \cdots, k < \nu$:

- $v_n \in K_{n+1}$
  $w_m \in K_{m+1}$
- $< w_m, v_n >_B = \begin{cases} 0 & \text{si } m \neq n \\ \delta_n & \text{si } m = n \end{cases}$

Montrons que ces deux propriétés sont également valables pour $n = k + 1$:

- $v_{k+1} \in K_{k+1}$ et $w_{k+1} \in \tilde{K}_{k+1}$ sont vérifiés par (10) et (5)

- $< w_m, v_{k+1} >= 0$ avec $m = 0, \cdots, k$ se montre comme suit :

$$\begin{aligned} < w_m, v_{k+1} > &= < w_m, \frac{Av_k - v_k\alpha_k - v_{k-1}\beta_{k-1}}{\gamma_k} >_B \\ &= (< w_m, Av_k >_B - < w_m, v_k\alpha_k >_B \\ &\quad - < w_m, v_{k-1}\beta_{k-1} >_B)/\gamma_k \end{aligned} \tag{1.32}$$

Or $< w_m, Av_k >_B = < A^* w_m, v_k >_B$ par (1.2)
et en remplaçant $A^* w_m$ par (5), nous avons :

$$\begin{aligned} < w_m, v_{k+1} > &= (< \tilde{\gamma}_m w_{m+1} + \tilde{\alpha}_m w_m + \tilde{\beta}_{m-1} w_{m-1}, v_k >_B - < w_m, v_k\alpha_k >_B \\ &\quad - < w_m, v_{k-1}\beta_{k-1} >_B)/\gamma_k \end{aligned} \tag{1.33}$$

* **pour** $m \leq k - 2$, nous obtenons $< w_m, v_{k+1} >= 0$
car la relation (1.33) peut se simplifier en sachant que $\tilde{K}_k \perp v_k$ par la proposition 1.1.1.

* **pour** $m = k-1$, nous obtenons $< w_{k-1}, v_{k+1} >_B = 0$
car en utilisant (1.25), la proposition 1.1.1 et la définition de δ_{k-1} dans l'équation (1.32),
nous avons $< w_{k-1}, v_{k+1} >= (\beta_{k-1}\delta_{k-1} - 0 - \beta_{k-1}\delta_{k-1}) = 0$.

* **pour** $m = k$, nous vérifions que $< w_k, v_{k+1} >_B = 0$
en utilisant la proposition 1.1.1 et la définition de δ_{k-1} dans (1.32),
$< w_k, v_{k+1} >= \frac{\alpha_k \delta_k - \alpha_k \delta_k}{\gamma_k} = 0$

- $< w_{k+1}, v_{k+1} >_B = \delta_{k+1}$
par (6) à (11).

□

1.4.2 Cas particulier : A est symétrique ou hermitienne

L'algorithme de biorthogonalisation peut se simplifier lorsque la matrice A est hermitienne ou symétrique complexe. Nous avons les résultats suivants :

- si A et B sont hermitiennes et B est définie positive

Les choix suivants :

1.) $v_0 = w_0$
2.) $\tilde{\gamma}_n = \gamma_n$
3.) $\delta_n > 0$

entraînent $\tilde{T}_\nu = \overline{T}_\nu = T_\nu$ et $v_n = w_n \forall n$.

Donc il n'est plus utile de calculer w_n et le coût est donc réduit de moitié. De plus, nous pouvons prendre $\gamma_n > 0\ \forall n$ ce qui signifie que $\beta_{n-1} > 0$. Finallement, nous pouvons choisir $\delta_n = \delta_0\ \forall n$, ce qui rend T_ν symétrique réel. Nous obtenons ainsi l'**algorithme de Lanczos symétrique** souvent simplement appelé algorithme de Lanczos.

- si A est symérique complexe

Le choix $w_0 = \overline{v}_0$ entraîne que $w_n = \overline{v}_n\ \forall n$. Il n'est donc plus nécessaire de calculer les deux suites, et si nous faisons le choix de $\delta_n = 1\ \forall n$, T_ν devient une matrice symétrique complexe.

1.4.3 " Les breakdowns"

Lors de l'application de l'algorithme de biorthogonalisation, un "breakdown" peut se produire. Il provoque l'arrêt prématuré du processus. Il existe deux types de "breakdowns" :
les "breakdowns prématurés" et "les breakdowns sérieux".

Arrêt prématuré

Il se produit quand :

$$v_\nu = 0 \ \ ou \ \ w_\nu = 0.$$

Nous allons tout d'abord introduire les notions de polynôme et de degré minimal d'une matrice.

Soit la séquence de matrices $\{I, A, A^2, \cdots, A^s\}$ telle que :

$$A^s + m_{s-1}A^{s-1} + \cdots + m_0 I = 0$$

où s est **le degré minimal** pour lequel la séquence de matrices $\{I, A, A^2, \cdots, A^{s-1}\}$ est linéairement indépendante.
Le polynôme m(λ) correspondant au membre de gauche de l'égalité ci-dessus est appelé **polynôme minimal de la matrice A**.

Proposition 1.4.1
Si $v_\nu = 0$ *, alors* $\{v_0, \cdots, v_{\nu-1}\}$ *engendrent un sous-espace A-invariant*
Si $w_\nu = 0$ *, alors* $\{w_0, \cdots, w_{\nu-1}\}$ *engendrent un sous-espace* A^**-invariant*

Preuve :
Si $v_\nu = 0$ dans la relation (1.12), nous obtenons $AV_\nu = V_\nu T_\nu$

□

Proposition 1.4.2
La dimension du sous-espace de Krylov $K_\nu(A, v_0)$ est au plus égale au degré du polynôme minimal de A.
Il en est de même pour le sous-espace $K_\nu(A^*, w_0)$.

Preuve :
Soit $p_m(A)$, le polynôme minimale de A, i.e. $p_m(A) = \sum_{j=0}^{m} \alpha_j A^j$ avec $\alpha_m = 1$ tel que $p_m(A) = 0$(cfr [8]). Nous avons alors $A^m = -\sum_{j=0}^{m-1} \alpha_j A^j$ ou encore, $A^m v_0 = -\sum_{j=0}^{m-1} \alpha_j A^j v_0$, de sorte que $K_m = K_{m-1}$.

□

Cette propriété nous montre que l'arrêt prématuré se produira au plus tard quand ν vaudra le degré du polynôme minimal de A.

"Breakdown sérieux"

Il se produit quand :

$$v_\nu \neq 0 \ \ et \ \ w_\nu \neq 0 \ \ mais \ \ < w_\nu, v_\nu >= 0. \tag{1.34}$$

Ce deuxième type de breakdown se produit quand les v_ν et w_ν sont orthogonaux l'un à l'autre, alors que $v_0 \neq w_0$. Ainsi, comme $< w_n, v_n >= 0$, nous ne vérifions plus la condition (1.4).

Il existe des techniques pour traiter ces breakdowns qui sont présentées par Gutknecht [7], par exemple.

Chapitre 2

L'algorithme QMR pour la résolution d'un système d'équations linéaires

2.1 Principe général de l'algorithme QMR

Soit à résoudre le système linéaire :

$$Ax = b \tag{2.1}$$

où $A \in \mathbb{C}^{N \times N}$ est une matrice non singulière.
L'itéré initial est noté x_0 et le $n^{ème}$ itéré x_n. Le résidu correspondant est défini par

$$r_n = b - Ax_n \tag{2.2}$$

Considérons la suite $v_0, v_1, \cdots, v_{n-1}$ de vecteurs de Lanczos à droite obtenus par l'algorithme de biorthogonalisation décrit au Chapitre 1, où la matrice A est celle du système (2.1), la matrice $B = I$ (ainsi nous travaillons avec la norme euclidienne), et où le vecteur $v_0 = r_0/||r_0||$ et w_0 est un vecteur arbitraire de $\mathbb{C}^N$, non orthogonal à v_0.
Le $n^{ème}$ itéré x_n est choisi de telle façon que

$$x_n \in x_0 + K_n(A, r_0)$$

ce qui s'écrit :

$$x_n = x_0 + V_n k_n \tag{2.3}$$

où $k_n \in \mathbb{C}^n$, $V_n = [v_0 \ v_1 \cdots v_{n-1}] \in \mathbb{C}^{N\times n}$.

Le résidu r_n s'écrit alors

$$\begin{aligned} r_n &= b - A(x_0 + V_n k_n) \\ &= r_0 - AV_n k_n \end{aligned} \tag{2.4}$$

Nous avons montré au chapitre 1 que $AV_n = V_{n+1}\underline{T_n}$, avec $\underline{T_n}$ défini en (1.20). Par ailleurs, $r_0 = v_0||r_0|| = V_{n+1}e_1||r_0|| = V_{n+1}e_1\rho_0$, où

$$\rho_0 = ||r_0||. \tag{2.5}$$

En posant :

$$d^{(n)} = e_1\rho_0 \in \mathbb{R}^{n+1}, \tag{2.6}$$

nous obtenons

$$\begin{aligned} r_n &= V_{n+1}d^{(n)} - V_{n+1}\underline{T_n}k_n \\ &= V_{n+1}(d^{(n)} - \underline{T_n}k_n) \end{aligned} \tag{2.7}$$

Idéalement, il serait souhaitable de déterminer le vecteur k_n de telle façon que la norme du vecteur r_n soit minimale. Le carré de cette norme s'écrit en utilisant le produit scalaire défini en (1.1) et l'égalité (2.7) :

$$||r_n||^2 = (d^{(n)} - \underline{T_n}k_n)^* V_{n+1}^* V_{n+1}(d^{(n)} - \underline{T_n}k_n)$$

Si les vecteurs $v_0, \cdots, v_{n-1}$ sont orthonormés, la norme $||r_n||^2$ se réduit à

$$||r_n||^2 = ||d^{(n)} - \underline{T_n}k_n||^2$$

En général, les vecteurs $v_0, \cdots, v_{n-1}$ ne constituent pas une suite orthonormée, de telle sorte que la détermination du vecteur k_n
minimisant $||r_n||^2$ est très coûteuse.

R.W. Freund et N.M. Nachtigal [4] ont proposé de remplacer le problème :

$$\min_{k_n \in \mathbb{C}^n} ||r_n||^2$$

par le problème :

$$\min_{k_n \in \mathbb{C}^n} ||d^{(n)} - \underline{T_n} k_n||^2, \tag{2.8}$$

ce qui revient à chercher la solution au sens des moindres carrés du système linéaire $\underline{T_n} k_n = d^{(n)}$.
La minimisation de la norme du résidu est donc remplacée par une " **Q**uasi **M**inimisation du **R**ésidu"(QMR).
La version de base de cet algorithme sans procédé de "look-ahead" peut se résumer comme suit :

étape 0

choix de x_0 et calcul de $r_0 = b - Ax_0$

étape 1

a) Première étape de l'algorithme de biorthogonalisation :

- choix de $v_0 = \dfrac{r_0}{||r_0||}$ et de w_0 tq $||w_0|| = 1$
- détermination de $\underline{T_1} \in \mathbb{C}^2$

b) Déterminer $x_1 = x_0 + k_1 v_0$ en résolvant $\min_{k_1 \in \mathbb{C}} ||d^{(1)} - \underline{T_1} k_1||^2$
c) Test d'arrêt :

- calcul de r_1
- Si x_1 satisfait à la condition d'arrêt, nous nous arrêtons, sinon, nous passons à l'étape 2.

étape 2
a) Deuxième étape de l'algorithme de biorthogonalisation :

- calcul de v_1 et de w_1
- détermination de $\underline{T_2} \in \mathbb{C}^{3\times 2}$

b) Déterminer $x_1 = x_0 + k_{1,1}v_0 + k_{2,1}v_1$ en résolvant $\min_{k_2 \in \mathbb{C}^2} ||d^{(2)} - \underline{T_2}k_2||^2$
où $k_2 = (k_{1,1}, k_{2,1})^T$
c) Test d'arrêt :

- calcul de r_2
- Si x_2 satisfait à la condition d'arrêt, nous nous arrêtons, sinon, nous passons à l'étape 3, et ainsi de suite...

2.2 Résolution du problème aux sens des moindres carrés

Ce problème peut être résolu grâce à la factorisation QR de la matrice $\underline{T_n}$. Le caractère tridiagonal de celle-ci permet de calculer cette factorisation en n rotations de Givens.

Rappelons que

$$\underline{T_n} = \begin{pmatrix} \alpha_0 & \beta_0 & 0 & \cdots & 0 \\ \gamma_0 & \alpha_1 & \beta_1 & \ddots & 0 \\ 0 & \gamma_1 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & \gamma_{n-2} & \alpha_{n-1} \\ 0 & 0 & 0 & 0 & \gamma_{n-1} \end{pmatrix}$$

où $\gamma_i \neq 0 \ (i = 0, \cdots, n-1)$.

Donc, la matrice $\underline{T_n}$ est de rang n.

Prenons :

$$\underbrace{\underline{T_n}}_{(n+1)\times n} = \underbrace{Q_n}_{(n+1)\times(n+1)} \underbrace{\underline{R_n}}_{(n+1)\times n} \tag{2.9}$$

comme décomposition QR de la matrice $\underline{T_n}$.

En fait, la dernière ligne de $\underline{R_n}$ est égale à zéro :

$$\underline{T_n} = Q_n \begin{pmatrix} R_n \\ 0 \end{pmatrix} \tag{2.10}$$

avec R_n matrice $n \times n$ et Q_n unitaire.

Définissons

$$\underline{h_n} = \begin{pmatrix} h_n \\ \hline \tilde{\eta}_{n+1} \end{pmatrix} = Q_n^* d^{(n)} \tag{2.11}$$

où $d^{(n)}$ est défini en (2.6).
Le problème de minimisation (2.8) peut être reformulé en considérant les égalités suivantes :

$$\begin{aligned} ||d^{(n)} - \underline{T_n} k_n||^2 &= ||Q_n^* d^{(n)} - \underline{R_n} k_n||^2 \text{ en multipliant par } Q_n^* \text{, et avec } Q_n \text{ unitaire} \\ &= ||\underline{h_n} - \underline{R_n} k_n||^2 \quad par(2.11) \\ &= ||h_n - R_n k_n||^2 + |\tilde{\eta}_{n+1}|^2 \end{aligned}$$

$|\tilde{\eta}_{n+1}|^2$ étant une constante, nous avons que

$$k_n, \text{solution de } \min_{k_n \in \mathbb{C}^n} ||d^{(n)} - \underline{T_n} k_n||^2, \tag{2.12}$$

est obtenu en résolvant

$$\min_{k_n \in \mathbb{C}^n} ||h_n - R_n k_n||^2, \tag{2.13}$$

c'est-à-dire en calculant la solution du système linéaire d'orde n :$R_n k_n = h_n$, dont la matrice R_n est triangulaire et de rang n. Cette solution existe et est unique.

Nous pouvons définir k_n :

$$k_n = R_n^{-1} h_n \tag{2.14}$$

Ainsi pour k_n solution de (2.12) et (2.13), nous avons

$$||d^{(n)} - \underline{T_n} k_n||^2 = |\tilde{\eta}_{n+1}|^2 \tag{2.15}$$

Pour connaître $||d^{(n)} - \underline{T_n} k_n||^2$, avec k_n solution de (2.12) et (2.13), il suffit de connaître la dernière composante de $Q_n^* d^{(n)}$.

2.3 Factorisation QR

2.3.1 Préliminaires

Nous définissons une **matrice de rotation** comme :

$$\begin{pmatrix} c_n & -s_n \\ \overline{s}_n & c_n \end{pmatrix} \tag{2.16}$$

avec $c_n \in I\!R^{\cancel{n}}$ et $c_n \geq 0$.

Il s'agit d'une matrice unitaire, dont le déterminant est égal à un, c'est-à-dire que $c_n^2 + |s_n|^2 = 1$.
Le but de cette matrice de rotation est, qu'une fois appliquée à un vecteur, elle permet de créer un zéro à un endroit précis. Ici nous désirons que

$$\begin{pmatrix} c_n & -s_n \\ \overline{s}_n & c_n \end{pmatrix}^* \begin{pmatrix} \mu_n \\ \nu_n \end{pmatrix} = \begin{pmatrix} \times \\ \mathbf{0} \end{pmatrix} \tag{2.17}$$

Cela doit donner

- si $\mu_n \neq 0$

  1.

$$\begin{aligned} \begin{pmatrix} c_n & s_n \\ -\overline{s}_n & c_n \end{pmatrix} \begin{pmatrix} \mu_n \\ \nu_n \end{pmatrix} &= \begin{pmatrix} \times \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} c_n\mu_n + s_n\nu_n \\ -\overline{s_n}\mu_n + c_n\nu_n \end{pmatrix} \end{aligned}$$

  Il faut donc que $-\overline{s_n}\mu_n + c_n\nu_n = 0$, et nous pouvons ainsi déterminer s_n :

$$s_n = c_n \overline{\frac{\nu_n}{\mu_n}} \tag{2.18}$$

  2. le déterminant doit être égal à un :

$$\begin{aligned} c_n^2 + |s_n|^2 &= 1 \\ c_n^2 + c_n^2 \frac{|\nu_n|^2}{|\mu_n|^2} &= 1 \\ c_n^2 \left(\frac{|\mu_n|^2 + |\nu_n|^2}{|\mu_n|^2}\right) &= 1 \end{aligned}$$

Nous obtenons donc :

$$c_n = \sqrt{\frac{|\mu_n|^2}{|\mu_n|^2 + |\nu_n|^2}} \tag{2.19}$$

• si $\mu_n = 0$

$$\begin{pmatrix} c_n & -s_n \\ \overline{s}_n & c_n \end{pmatrix} \begin{pmatrix} 0 \\ \nu_n \end{pmatrix} = \begin{pmatrix} \times \\ \mathbf{0} \end{pmatrix}$$

et donc

$$\begin{array}{l} c_n \nu_n = 0 \\ c_n^2 + |s_n|^2 = 1 \end{array} \Longrightarrow \text{une solution est } \begin{array}{l} s_n = 1 \\ c_n = 0 \end{array}$$

2.3.2 Décomposition QR

Nous allons regarder comment, étape par étape, la factorisation QR se construit.
Nous utiliserons la matrice de rotation décrite au paragraphe précédent.

$\boxed{\text{factorisation de } \underline{T_1}}$

$\underline{T_1} = \begin{pmatrix} \alpha_0 \\ \gamma_0 \end{pmatrix}$

Nous effectuons $Q_1^* \underline{T_1} = \underline{R_1}$

$$\Longrightarrow \begin{pmatrix} c_1 & -s_1 \\ \overline{s}_1 & c_1 \end{pmatrix}^* \begin{pmatrix} \alpha_0 \\ \gamma_0 \end{pmatrix} = \begin{pmatrix} \times \\ \mathbf{0} \end{pmatrix}$$

Ainsi nous obtenons bien $\underline{T_1} = Q_1 \underline{R_1}$.

$\boxed{\text{factorisation de } \underline{T_2}}$

$$\underline{T_2} = \begin{pmatrix} \alpha_0 & \beta_0 \\ \gamma_0 & \alpha_1 \\ 0 & \gamma_1 \end{pmatrix}$$

• Si nous voulons préserver la structure d'une matrice triangulaire supérieure déjà établie, nous faisons :

$$\left(\begin{array}{c|c} Q_1^* & 0 \\ \hline 0 & 1 \end{array}\right) \begin{pmatrix} \alpha_0 & \beta_0 \\ \gamma_0 & \alpha_1 \\ 0 & \gamma_1 \end{pmatrix} = \begin{pmatrix} \times & \times \\ 0 & \times \\ 0 & \gamma_1 \end{pmatrix}$$

• Nous voulons ensuite continuer à engendrer la structure de la matrice triangulaire supérieur R_2, en annulant l'élément (3,2) du second membre de l'équation ci-dessus.

Nous allons prémultiplier par $G_2^* = \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & c_2 & s_2 \\ 0 & -\bar{s}_2 & c_2 \end{array}\right)$

tel que $\begin{pmatrix} c_2 & s_2 \\ -\bar{s}_2 & c_2 \end{pmatrix} \begin{pmatrix} \times \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} \times \\ \mathbf{0} \end{pmatrix}$

Nous obtenons donc

$$\underbrace{\left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & c_2 & s_2 \\ 0 & -\bar{s}_2 & c_2 \end{array}\right) \left(\begin{array}{c|c} Q_1^* & 0 \\ \hline 0 & 1 \end{array}\right)}_{Q_2^*} \underbrace{\begin{pmatrix} \alpha_0 & \beta_0 \\ \gamma_0 & \alpha_1 \\ 0 & \gamma_1 \end{pmatrix}}_{\underline{T_2}} = \underbrace{\begin{pmatrix} \times & \times \\ 0 & \times \\ 0 & \mathbf{0} \end{pmatrix}}_{\underline{R_2}}$$

avec $Q_2 = \left(\begin{array}{c|c} Q_1 & 0 \\ \hline 0 & 1 \end{array}\right) \underbrace{\left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & c_2 & -s_2 \\ 0 & \bar{s}_2 & c_2 \end{array}\right)}_{G_2}$

Nous obtenons $\underline{T_2} = Q_2 \underline{R_2}$

$\boxed{\text{factorisation de } \underline{T_n}}$

La matrice Q_n est un produit de n rotations de Givens choisies pour annuler les éléments sous-diagonaux de $\underline{T_n}$. Il suffira de connaître la forme factorisée de Q_n.

Soit Q_n matrice $(n+1) \times (n+1)$
G_n matrice $(n+1) \times (n+1)$

$$Q_n = \left(\begin{array}{c|c} Q_{n-1} & 0 \\ \hline 0^T & 1 \end{array} \right) G_n \text{ avec } G_n = \left(\begin{array}{c|cc} I_{n-1} & 0 & 0 \\ \hline 0^T & c_n & -s_n \\ 0^T & \overline{s}_n & c_n \end{array} \right) \tag{2.20}$$

avec $c_n \geq 0$ et $s_n \in \mathbb{C}^n$ tel que $c_n^2 + |s_n|^2 = 1$
Nous choisissons c_n et s_n tels que

$$G_n^* \begin{pmatrix} * \\ \vdots \\ * \\ \mu_n \\ \nu_n \end{pmatrix} = \begin{pmatrix} * \\ \vdots \\ * \\ c_n\mu_n + s_n\nu_n \\ 0 \end{pmatrix} avec \begin{pmatrix} * \\ \vdots \\ * \\ \mu_n \\ \nu_n \end{pmatrix} = \left(\begin{array}{c|c} Q_{n-1}^* & 0 \\ \hline 0^T & 1 \end{array} \right) \underline{T_n} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{2.21}$$

Nous avons, comme nous l'avons vu dans les préliminaires,

- si $\mu_n \neq 0$: $\quad c_n = \frac{|\mu_n|}{\sqrt{|\mu_n|^2+|\nu_n|^2}} \quad s_n = \overline{c_n \frac{\nu_n}{\mu_n}}$
- si $\mu_n = 0$: $\quad c_n = 0 \quad s_n = 1$

2.4 Calcul des itérés

Nous supposons que nous connaissons x_{n-1}.
Nous allons d'abord expliciter h_n, donné par (2.11), à partir de h_{n-1} et ainsi pouvoir construire x_n en fonction de h_n et x_{n-1}.

2.4.1 Mise à jour de h_n

Nous connaissons h_{n-1} : $\underline{h}_{n-1} = \begin{pmatrix} h_{n-1} \\ \hline \tilde{\eta}_n \end{pmatrix} = Q^*_{n-1} d^{(n-1)} \in \mathbb{C}^n$
Ainsi la formule de mise à jour de $\underline{h_n}$ est simplifiée :

$$\begin{aligned}
\underline{h_n} &= \begin{pmatrix} h_n \\ \hline \tilde{\eta}_{n+1} \end{pmatrix} = Q^*_n d^{(n)} && (2.22) \\
&= \left(\left(\begin{array}{c|c} Q_{n-1} & 0 \\ \hline 0^T & 1 \end{array} \right) G_n \right)^* d^{(n)} \\
&= G^*_n \left(\begin{array}{c|c} Q^*_{n-1} & 0 \\ \hline 0^T & 1 \end{array} \right) \begin{pmatrix} \rho_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
&= G^*_n \begin{pmatrix} Q^*_{n-1} d^{(n-1)} \\ \hline 0 \end{pmatrix} \\
&= G^*_n \begin{pmatrix} \underline{h}_{n-1} \\ \hline 0 \end{pmatrix} \\
&= G^*_n \begin{pmatrix} h_{n-1} \\ \hline \tilde{\eta}_n \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} h_{n-1} \\ \hline c_n \tilde{\eta}_n \\ -\overline{s}_n \tilde{\eta}_n \end{pmatrix} && (2.23)
\end{aligned}$$

Ainsi

$$\left(\; h_n \; \right) = \begin{pmatrix} h_{n-1} \\ \hline c_n \tilde{\eta}_n \end{pmatrix} \qquad (2.24)$$

$h_n \in \mathbb{C}^n$ est donc obtenu à partir de $h_{n-1} \in \mathbb{C}^{n-1}$ en lui "ajoutant" une dernière composante égale à $c_n \tilde{\eta}_n$.
Nous avons vu en (2.15) que si k_n est la solution du problème de minimisation (2.12), alors $\tilde{\eta}_{n+1} = ||d^{(n)} - \underline{T_n} k_n||$.

Nous voyons , en considérant (2.22) et (2.23) que :

$$||d^{(n)} - \underline{T_n} k_n|| = |\tilde{\eta}_{n+1}| = |s_n \tilde{\eta}_n| = |s_1 s_2 \cdots s_n| ||r_0|| \tag{2.25}$$

car comme $\tilde{\eta}_{n+1} = -\overline{s_n} \tilde{\eta}_n$, nous avons

$$\begin{aligned} |\tilde{\eta}_{n+1}| &= |s_n \tilde{\eta}_n| \\ |\tilde{\eta}_n| &= |s_{n-1} \tilde{\eta}_{n-1}| \\ |\tilde{\eta}_2| &= |s_1 \tilde{\eta}_1| \\ |\tilde{\eta}_1| &= |s_0 \tilde{\eta}_0| = ||r_0|| \end{aligned} \tag{2.26}$$

2.4.2 Calcul de l'itéré x_n

Le vecteur x_n est donné par (2.3).
En notant $Z_n := (z_0 \cdots z_{n-1}) := V_n(R_n)^{-1}$, nous obtenons $x_n = x_0 + Z_n h_n$.
Nous avons donc que $x_{n-1} = x_0 + [z_0 \cdots z_{n-2}] h_{n-1}$
et par (2.24), nous obtenons :

$$\begin{aligned} x_n &= x_0 + [z_0 \cdots z_{n-2} | z_{n-1}] \left(\frac{h_{n-1}}{c_n \tilde{\eta}_n} \right) \\ &= x_0 + [z_0 \cdots z_{n-2}] h_{n-1} + z_{n-1} \tilde{\eta}_n c_n \\ x_n &= x_{n-1} + z_{n-1} c_n \tilde{\eta}_n \end{aligned} \tag{2.27}$$

Ainsi pour calculer x_n, il nous faudra connaître $\tilde{\eta}_n$ et z_{n-1}. Le calcul de z_{n-1} est assez aisé vu que R_n est une matrice tridiagonale supérieure. Nous obtenons donc une récurrence à trois termes.

2.5 Calcul du vecteur résidu

Regardons si nous pouvons réécrire le résidu à l'aide du calcul de h_n :

$$
\begin{aligned}
r_n &= V_{n+1}(d^{(n)} - \underline{T_n}k_n) \qquad par(2.7) \\
&= V_{n+1}(d^{(n)} - \underline{T_n}(R_n)^{-1}h_n) \qquad par(2.14) \\
&= V_{n+1}(d^{(n)} - Q_n\underline{R_n}(R_n)^{-1}h_n) \qquad par(2.9) \\
&= V_{n+1}(d^{(n)} - Q_n \left(\frac{h_n}{0}\right)) \\
&= V_{n+1}Q_n\underline{h_n} - V_{n+1}Q_n \left(\frac{h_n}{0}\right) par(2.11) \\
&= V_{n+1}Q_n \begin{pmatrix} 0 \\ \tilde{\eta}_{n+1} \end{pmatrix} \\
&= V_{n+1}Q_n e_{n+1}\tilde{\eta}_{n+1} \qquad (2.28)
\end{aligned}
$$

ainsi

$$
\begin{aligned}
r_n &= \left(V_n \mid v_n \right) \left(\begin{array}{c|c} Q_{n-1} & 0 \\ \hline 0^T & 1 \end{array}\right) G_n \left(\frac{0}{\tilde{\eta}_{n+1}}\right) \\
&= \left(V_nQ_{n-1} \mid v_n \right) \left(\begin{array}{c|cc} I_{n-1} & 0 & 0 \\ \hline 0 & c_n & -s_n \\ 0 & \overline{s}_n & c_n \end{array}\right) \left(\begin{array}{c} 0 \\ 0 \\ \hline \tilde{\eta}_{n+1} \end{array}\right) \\
&= V_nQ_{n-1} \left(\begin{array}{c} 0 \\ 0 \\ \hline -s_n\tilde{\eta}_{n+1} \end{array}\right) + v_nc_n\tilde{\eta}_{n+1} \\
&= -V_nQ_{n-1}e_ns_n\tilde{\eta}_{n+1} + v_nc_n\tilde{\eta}_{n+1} \\
&= V_nQ_{n-1}e_n|s_n|^2\tilde{\eta}_n + v_nc_n\tilde{\eta}_{n+1} \qquad par(2.26) \\
&= r_{n-1}|s_n|^2 + v_nc_n\tilde{\eta}_{n+1} \qquad par(2.28) \qquad (2.29)
\end{aligned}
$$

2.6 Algorithme QMR

1. <u>étape 0</u>
   - choisir x_0
   - calcul $r_0 = b - Ax_0$
   - $n = 1$
2. <u>étape n</u>
   - $n-1^{\text{ème}}$ étape de l'algorithme de Biorthogonalisation
   - factorisation QR

   (a) <u>calcul de μ_n, ν_n tel que</u>

$$\begin{pmatrix} * \\ \vdots \\ * \\ \mu_n \\ \nu_n \end{pmatrix} = \left(\begin{array}{c|c} Q^*_{n-1} & 0 \\ \hline 0^T & 1 \end{array}\right) \underline{T_n} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

   (b) <u>calcul de c_n, s_n</u>
   - si $\mu_n \neq 0$: $\qquad c_n = \frac{|\mu_n|}{\sqrt{|\mu_n|^2+|\nu_n|^2}} \qquad s_n = \overline{c_n \frac{\nu_n}{\mu_n}}$
   - si $\mu_n = 0$: $\qquad c_n = 0 \qquad s_n = 1$

   (c) <u>calcul de G_n</u>

$$G_n = \left(\begin{array}{c|cc} I_{n-1} & 0 & 0 \\ \hline 0^T & c_n & -s_n \\ 0^T & \overline{s}_n & c_n \end{array}\right)$$

   (d) <u>calcul de Q_n</u>

$$Q_n = \left(\begin{array}{c|c} Q_{n-1} & 0 \\ \hline 0^T & 1 \end{array}\right) G_n$$

   - calcul de h_n et $\tilde{\eta}_{n+1}$

$$\underline{h_n} = \begin{pmatrix} h_n \\ \hline \tilde{\eta}_{n+1} \end{pmatrix} = \begin{pmatrix} h_{n-1} \\ \hline c_n\tilde{\eta}_n \\ -\overline{s}_n\tilde{\eta}_n \end{pmatrix}$$

   c'est-à-dire

$$h_n = \begin{pmatrix} h_{n-1} \\ \hline c_n\tilde{\eta}_n \end{pmatrix}$$

$$\tilde{\eta}_{n+1} = -\overline{s}_n\tilde{\eta}_n$$

– calcul de z_{n-1}

$$V_n = R_n Z_n$$

– calcul de x_n et r_n

$$x_n = x_{n-1} + z_{n-1} c_n \tilde{\eta}_n$$

$$r_n = r_{n-1} |s_n|^2 + v_n c_n \tilde{\eta}_{n+1}$$

– condition d'arrêt

IF $\frac{||r_n||}{||r_0||} < \epsilon$

THEN stop

ELSE $n = n + 1$

Chapitre 3

Une variante de l'algorithme QMR

La méthode QMR pour la résolution des systèmes linéaires a été proposée en 1991 par R.W. Freund et N.M. Nachtigal [4]. Elle consiste à exprimer les itérés successifs dans une base constituée des vecteurs de Lanczos. Ceux-ci sont obtenus par une récurrence à trois termes (1.10), et chaque pas de l'algorithme conduit à un problème de minimisation au sens des moindres carrés (2.8), résolu par une factorisation QR.

En 1994, ces auteurs [5] ont proposé une autre implémentation de la méthode QMR où les vecteurs de Lanczos sont obtenus par un couple de récurrences à deux termes.

Plus récemment, M. Sauren et H.M. Bucker [12] ont utilisé cette nouvelle manière d'obtenir des vecteurs de Lanczos et ont évité la factorisation QR lors du problème de minimisation au sens des moindres carrés (2.8). Ils proposent de résoudre celui-ci grâce à une récurrence simple. Le présent chapitre est consacré à l'analyse de leur démarche.

3.1 Préliminaires

R.W. Freund et N.M. Nachtigal [5] décrivent une autre façon de construire l'algorithme de Lanczos. En fait, ils remplacent la récurrence à trois termes (1.10) par un couple de récurrences à deux termes. Ils considérent les deux suites finies de vecteurs $\{v_j\}_{j=0}^{\nu-1}$ et $\{w_j\}_{j=0}^{\nu-1}$ décrites au chapitre 1 et vérifiant les conditions (1.4) et (1.3) avec $B = I$. Ils initialisent v_0 et w_0 comme pour le chapitre 2, c'est-à-dire : $v_0 = r_0/||r_0||_2$ et w_0 est un vecteur arbitraire de $\mathbb{C}^N$, non orthogonal à v_0.
De plus ils travaillent avec une nouvelle paire de suites $\{p_j\}_{j=0}^{\nu-1}$ et $\{q_j\}_{j=0}^{\nu-1}$ pour générer les vecteurs v_j et w_j.

Ces nouvelles suites satisfont aux conditions suivantes :

- $$p_j \in K_{j+1}(A, v_0) = span\{v_0, Av_0, ..., A^j v_0\}$$
  $$q_j \in K_{j+1}(A^*, w_0) = span\{w_0, A^* w_0, ..., A^{*^j} w_0\} \tag{3.1}$$
- $$q_j^T A p_i = \begin{cases} 0 & \text{si } j \neq i \\ \epsilon_i \neq 0 & \text{si } j = i \end{cases} \tag{3.2}$$

Pour $n \leq \nu$, le lien entre les vecteurs $\{v_j\}_{j=0}^{n}$ et $\{p_j\}_{j=0}^{n-1}$ est donné par ces relations :

$$V_n = P_n U_n \tag{3.3}$$
$$AP_n = V_{n+1} L_n \tag{3.4}$$

où

$$P_n := [p_0 \cdots p_{n-1}] \text{ et } V_n := [v_0 \cdots v_{n-1}] \tag{3.5}$$

$$U_n = \begin{pmatrix} 1 & u_{0,1} & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & \ddots & u_{(n-2),(n-1)} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{3.6}$$

$$L_n = \begin{pmatrix} \tau_0 & 0 & \cdots & 0 \\ \rho_1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & \ddots & \tau_{n-1} \\ 0 & 0 & \cdots & \rho_n \end{pmatrix} \tag{3.7}$$

L'algorithme de Lanczos basé sur un couple de récurrences à deux termes ne sera pas détaillé ici. Le lecteur pourra trouver les détails de son implémentation dans [5]. Retenons cependant que cet algorithme fournit les vecteurs p_i et v_i, ainsi que les éléments des matrices L et U, c'est-à-dire en particulier τ_i, ρ_{i+1} pour $i = 0, \cdots, n-1$.

Nous pouvons établir une relation entre $\underline{T_n}$ défini en (1.20) et $L_n U_n$:

$$AV_n = V_{n+1} L_n U_n \quad par \ (3.3) \ et \ (3.4)$$
$$Or \ \ AV_n = V_{n+1} \underline{T_n} \quad par (1.18)$$

Par conséquent,

$$\underline{T_n} = L_n U_n \tag{3.8}$$

De plus, nous pouvons remarquer que L_n est de rang plein car d'une part $rg(L_n) \leq n$ et de l'autre $rg(\underline{T_n}) = n \leq rg(L_n)$.

3.2 Calcul des itérés

Soit à résoudre le système linéaire

$$Ax = b \tag{3.9}$$

où $A \in \mathbb{C}^{N \times N}$ est une matrice non singulière.

L'itéré initial est noté x_0 et le $n^{ème}$ itéré x_n. Le résidu correspondant est défini par

$$r_n = b - Ax_n \tag{3.10}$$

Considérons la suite $v_0, v_1, \cdots, v_{n-1}$ de vecteurs de Lanczos à droite obtenue par l'algorithme de biorthogonalisation, où la matrice A est celle du système (3.9), la matrice $B = I$, le vecteur $v_0 = r_0/||r_0||_2$ et w_0 est un vecteur arbitraire de $\mathbb{C}^N$, non orthogonal à v_0.
Le $n^{ème}$ itéré x_n est choisi de telle façon que

$$x_n \in x_0 + K_n(A, r_0)$$

ce qui s'écrit :

$$x_n = x_0 + V_n k_n \tag{3.11}$$

où $k_n \in \mathbb{C}^n$, $V_n = [v_0 \;\; v_1 \cdots v_{n-1}] \in \mathbb{C}^{N \times n}$.

Par (3.3), nous pouvons réécrire (3.11) :

$$\begin{aligned} x_n &= x_0 + P_n U_n k_n \\ &= x_0 + P_n y_n \end{aligned} \tag{3.12}$$

où P_n est défini en (3.5),
et en posant :

$$y_n := U_n k_n \tag{3.13}$$

Le résidu r_n s'écrit alors

$$\begin{aligned} r_n &= b - Ax_n \\ &= b - Ax_0 - AP_n y_n \quad par\ (3.12) \\ &= r_0 - V_{n+1} L_n y_n \quad par\ (3.4) \\ &= V_{n+1} d^{(n)} - V_{n+1} L_n y_n \quad avec\, d^{(n)} \text{ défini par (2.6)} \\ r_n &= V_{n+1}(d^{(n)} - L_n y_n) \end{aligned} \tag{3.14}$$

avec la matrice L_n décrite en (3.7) et V_{n+1} défini en (3.5).
Tout comme au chapitre 2, idéalement il serait souhaitable de déterminer le vecteur y_n de telle façon que la norme du vecteur r_n soit minimale. Mais la minimisation est fort coûteuse. H. Sauren et H.M. Bucker [12] proposent de trouver la solution de

$$\min_{y \in \mathbb{C}^n} ||d^{(n)} - L_n y||_2 = ||d^{(n)} - L_n y_n||_2 \tag{3.15}$$

La méthode classique pour résoudre (3.15) est basée sur la décomposition QR de L_n. Mais ici nous suivrons une approche différente.

Lemme 3.2.1
Soit pour $n = 1, 2, \cdots,$
$L_n \in \mathbb{C}^{(n+1)\times n}$ *défini en (3.7),*
$y_n \in \mathbb{C}^n$ *étant l'unique solution du problème aux moindres carrés (3.15) pour* $\rho_0 \in \mathbb{C}$,
Alors le vecteur y_n *défini comme suit est la solution du problème aux moindres carrés :*

$$y_n = (1 + \theta_{n-1}) \begin{pmatrix} y_{n-1} \\ 0 \end{pmatrix} - \theta_{n-1} \begin{pmatrix} y_{n-2} \\ 0 \\ 0 \end{pmatrix} + \kappa_{n-1} e_n \quad pour\ n \geq 3 \tag{3.16}$$

avec
$y_1 \;= \kappa_0$
$y_2 \;= (\kappa_0 + \theta_0 \kappa_0, \kappa_1)^T$
où

$$\theta_j = \frac{|\tau_j|^2 (1 - \sigma_j)}{\sigma_j |\tau_j|^2 + |\rho_{j+1}|^2} \quad pour\ j \geq 0 \tag{3.17}$$

$$\kappa_j = \frac{-\rho_j \overline{\tau}_j \kappa_{j-1}}{\sigma_j |\tau_j|^2 + |\rho_{j+1}|^2} \quad pour\ j \geq 0\ et\ \kappa_{-1} = -1, \tag{3.18}$$

$$\sigma_{j+1} = \frac{\sigma_j |\tau_j|^2}{\sigma_j |\tau_j|^2 + |\rho_{j+1}|^2} \quad pour\ j \geq 0, \sigma_0 = 1 \tag{3.19}$$

et τ_j défini en (3.7).

<u>**Preuve**</u> :
Tout d'abord, nous avons que les θ_j, κ_j et σ_{j+1} sont bien définis par (3.17), (3.18),(3.19), puisque nous savons que la matrice L_n est de rang plein et que cela implique que quand les $\tau_i = 0$, alors les $\rho_j \neq 0$ pour $j = i+1, \cdots, n-1$.
Ainsi la récurrence (3.16) est également bien définie.
Par (3.16), nous pouvons remarquer que κ_{n-1} est la dernière composante de y_n.
Par calcul, nous obtenons :

$$M_n := L_n^* L_n = \begin{pmatrix} |\tau_0|^2 + |\rho_1|^2 & \overline{\rho}_1 \tau_1 & 0 & \cdots & 0 \\ \rho_1 \overline{\tau}_1 & |\tau_1|^2 + |\rho_2|^2 & \overline{\rho}_2 \tau_2 & 0 & 0 \\ 0 & \rho_2 \overline{\tau}_2 & \ddots & \ddots & \\ \vdots & \cdots & \ddots & \ddots & \overline{\rho}_{n-1} \tau_{n-1} \\ 0 & \cdots & 0 & \rho_{n-1} \overline{\tau}_{n-1} & |\tau_{n-1}|^2 + |\rho_n|^2 \end{pmatrix} \in \mathbb{C}^{n \times n}$$

Nous devons vérifier par induction que la définition de y_n par la récurrence (3.16) est bien la solution du problème aux moindres carrés (3.15). Pour cela y_n défini en (3.16) doit être solution des équations normales, c'est-à-dire :

$$M_n y_n = L_n^* L_n y_n = L_n^* d^{(n)} = \rho_0 \overline{\tau}_0 e_1 \tag{3.20}$$

La dernière égalité est vérifiée vu la structure de la matrice L_n^*.

Regardons si l'équation (3.20) est vérifiée pour $n = 1$ et $n = 2$
• <u>$n = 1$</u>

Par l'équation (3.16), nous obtenons :

$$y_1 = \frac{\overline{\tau_0} \rho_0}{|\tau_0|^2 + |\rho_1|^2} \tag{3.21}$$

Il faut vérifier que y_1 est la solution du problème $\min_{y \in \mathbb{C}} ||d^{(1)} - L_1 y||_2$, c'est-à-dire que y_1 doit vérifier l'équation normale :

$$\begin{aligned} \begin{pmatrix} \overline{\tau_0} & \overline{\rho_1} \end{pmatrix} \begin{pmatrix} \overline{\tau_0} \\ \overline{\rho_1} \end{pmatrix} y_1 &= \begin{pmatrix} \overline{\tau_0} & \overline{\rho_1} \end{pmatrix} \begin{pmatrix} \overline{\rho_0} \\ 0 \end{pmatrix} \\ &\Longleftrightarrow \\ |\tau_0|^2 + |\rho_1|^2 y_1 &= \overline{\tau_0} \rho_0 \\ &\Longleftrightarrow \\ y_1 &= \frac{\overline{\tau_0} \rho_0}{|\tau_0|^2 + |\rho_1|^2} \end{aligned} \tag{3.22}$$

Ainsi y_1 la solution de l'équation normale correspond à la définition de y_1 donnée en (3.16).

• $\underline{n = 2}$

Il faut résoudre un système à deux équations et deux inconnues. Nous trouvons bien, après calcul, y_2.

• $\underline{n \geq 3}$

Nous pouvons écrire à l'aide de l'équation (3.16) le produit de M_n par y_n :

$$M_n y_n = (1+\theta_{n-1}) M_n \begin{pmatrix} y_{n-1} \\ 0 \end{pmatrix} - \theta_{n-1} M_n \begin{pmatrix} y_{n-2} \\ 0 \\ 0 \end{pmatrix} + \kappa_{i-1} M_n e_n \tag{3.23}$$

Il faut donc que ce y_n soit solution des équations normales. Par induction, nous supposons que pour $i < n$:

$$\begin{aligned} M_i y_i &= \rho_0 \overline{\tau}_0 e_1 \\ \textit{et } q_i &= \kappa_{i-1}. \end{aligned} \tag{3.24}$$

Et nous devons montrer que y_i pour $i = n$ est bien solution du problème aux moindres carrés. Nous pouvons réécrire M_i en fonction de M_{i-1} et de M_{i-2} pour pouvoir ainsi réécrire (3.23) :

$$M_i = \left(\begin{array}{cccc|c} & & & & 0 \\ & & & & \vdots \\ & M_{i-1} & & & 0 \\ & & & & \tau_{i-1}\overline{\rho}_{i-1} \\ \hline 0 & \cdots & 0 & \overline{\tau}_{i-1}\rho_{i-1} & |\tau_{i-1}|^2 + |\rho_i|^2 \end{array}\right)$$

$$M_i = \left(\begin{array}{ccc|c|c} & M_{i-2} & & \begin{matrix} 0 \\ \vdots \\ \tau_{i-2}\overline{\rho}_{i-2} \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 & \cdots & \overline{\tau}_{i-2}\rho_{i-2} & |\tau_{i-2}|^2 + |\rho_{i-1}|^2 & \tau_{i-1}\overline{\rho}_{i-1} \\ \hline 0 & \cdots & 0 & \overline{\tau}_{i-1}\rho_{i-1} & |\tau_{i-1}|^2 + |\rho_i|^2 \end{array}\right)$$

Par ces deux égalités, et en notant q_i la dernière composante de y_i, nous obtenons :

$$M_i y_i = (1+\theta_{i-1}) \begin{pmatrix} M_{i-1} y_{i-1} \\ \rho_{i-1}\overline{\tau}_{i-1} q_{i-1} \end{pmatrix} - \theta_{i-1} \begin{pmatrix} M_{i-2} y_{i-2} \\ \rho_{i-2}\overline{\tau}_{i-2} q_{i-2} \\ 0 \end{pmatrix} + \kappa_{i-1} \begin{pmatrix} O^{(i-2)} \\ \overline{\rho}_{i-1}\tau_{i-1} \\ |\tau_{i-1}|^2 + |\rho_i|^2 \end{pmatrix}$$

En utilisant l'hypothèse d'induction (3.24), nous obtenons :

$$M_i y_i = \begin{pmatrix} \rho_0 \overline{\tau}_0 e_1^{(i-2)} \\ -\theta_{i-1}\rho_{i-2}\overline{\tau}_{i-2}\kappa_{i-2} + \kappa_{i-1}\overline{\rho}_{i-1}\tau_{i-1} \\ (1+\theta_{i-1})\rho_{i-1}\overline{\tau}_{i-1}\kappa_{i-2} + \kappa_{i-1}(|\tau_{i-1}|^2 + |\rho_i|^2) \end{pmatrix}$$

Or par les équations (3.17)-(3.19), nous avons que les deux dernières composantes du vecteur sont égales à zéro, ce qui entraîne que nous avons bien $M_i y_i = \rho_0 \overline{\tau}_0 e_1^{(i)}$ pour $i = n$.

□

Nous pouvons réécrire la récurrence (3.16) de la façon suivante :

Corollaire 3.2.1
Pour $n \geq 1$ et $y_n, g_n \in \mathbb{C}^n$, le couple d'itérations suivant donne la solution du problème aux moindres carrés (3.15) :

$$y_n = \begin{pmatrix} y_{n-1} \\ 0 \end{pmatrix} + g_n \quad \textit{pour } n \geq 2 \tag{3.25}$$

$$g_n = \theta_{n-1}\begin{pmatrix} g_{n-1} \\ 0 \end{pmatrix} + \kappa_{n-1} e_n \quad \textit{pour } n \geq 2 \tag{3.26}$$

avec $y_1 = g_1 = \kappa_0$, *θ_{n-1} défini en (3.17) et κ_{n-1} défini en (3.18).*

A l'aide de (3.25), l'équation(3.12) s'écrit comme suit :

$$\begin{aligned} x_n &= x_0 + P_{n-1}y_{n-1} + P_n g_n \\ &= x_{n-1} + P_n g_n \\ &= x_{n-1} + f_n \end{aligned} \tag{3.27}$$

avec

$$f_n = P_n g_n \tag{3.28}$$

Si nous utilisons de plus la récurrence (3.26), nous obtenons :

$$\begin{aligned} f_n &= \theta_{n-1}P_{n-1}g_{n-1} + \kappa_{n-1}P_n e_n \\ &= \theta_{n-1}f_{n-1} + \kappa_{n-1}p_{n-1} \end{aligned} \tag{3.29}$$

avec $f_0 = 0$, θ_{n-1} défini en (3.17), κ_{n-1} défini en (3.18), et où les vecteurs p_n sont construits par le procédé de Lanczos.

Si nous définissons $s_n = Af_n$, nous pouvons réécrire le résidu (3.10) à l'aide de (3.27) comme suit :

$$r_n = r_{n-1} - s_n \tag{3.30}$$

Nous pouvons également réécrire s_n à l'aide de f_n , défini en (3.29), de la manière suivante :

$$s_n = \theta_{n-1} s_{n-1} + \kappa_{n-1} A p_{n-1} \tag{3.31}$$

où $s_0 = 0$

3.3 Algorithme QMR

1. Initialisation

   - Choisir x_0

   - $f_0 = s_0 = 0, \sigma_0 = 1$ et $\kappa_{-1} = -1$

   - Calculer $r_0 = b - Ax_0$

   - Initialiser l'algorithme de Lanczos avec $\rho_0 = ||r_0||_2$ et $v_0 = \frac{r_0}{\rho_0}$

2. Etape n

   - Générer $p_{n-1}, Ap_{n-1}, \tau_{n-1}$ et ρ_n en appliquant le $n^{ème}$ pas de l'agorithme de Lanczos basé sur un couple de récurrences à deux termes.

   - Calculs

$$
\begin{aligned}
\theta_{n-1} &= \frac{|\tau_{n-1}|^2(1-\sigma_{n-1})}{\sigma_{n-1}|\tau_{n-1}|^2+|\rho_n|^2} \\
\kappa_{n-1} &= \frac{-\rho_{n-1}\overline{\tau}_{n-1}\kappa_{n-2}}{\sigma_{n-1}|\tau_{n-1}|^2+|\rho_n|^2} \\
\sigma_n &= \frac{\sigma_{n-1}|\tau_{n-1}|^2}{\sigma_{n-1}|\tau_{n-1}|^2+|\rho_n|^2} \\
f_n &= \theta_{n-1}f_{n-1} + \kappa_{n-1}p_{n-1} \\
s_n &= \theta_{n-1}s_{n-1} + \kappa_{n-1}Ap_{n-1} \\
x_n &= x_{n-1} + f_n \\
r_n &= r_{n-1} - s_n
\end{aligned}
$$

   - Condition d'arrêt

     IF $\frac{||r_n||}{||r_0||} < \epsilon$

     THEN stop

     ELSE $n = n + 1$

Chapitre 4

La méthode QMR appliquée à l'équation matricielle de Lyapunov

4.1 Introduction

Au chapitre 2, nous avons décrit la méthode QMR pour la résolution d'un système d'équations linéaires. Dans ce chapitre, nous allons montrer que cette méthode QMR peut également être utilisée pour la résolution de l'équation matricielle de Lyapunov.

L'**équation de Lyapunov** est définie par :

$$AX + XA^T = -D^T D \tag{4.1}$$
$$\text{où } A \in \mathbb{C}^{N\times N}, D \in \mathbb{C}^{N\times N}$$

Si nous définissons l'opérateur L appelé **opérateur de Lyapunov** par

$$L : \begin{cases} \mathbb{C}^{N\times N} & \longrightarrow \mathbb{C}^{N\times N} \\ X & \rightsquigarrow AX + XA^T \end{cases} \tag{4.2}$$

Nous pouvons réécrire l'équation (4.1) comme :

$$L(X) = F \quad avec \quad F = -D^T D \tag{4.3}$$

Notre démarche est la suivante : nous allons écrire l'équation matricielle (4.1) comme un système d'équations linéaires à N^2 inconnues qui sont les éléments de X. Et nous lui appliquerons la méthode QMR. Ensuite nous réécrirons l'algorithme QMR applicable directement à l'équation de Lyapunov.

• Nous allons définir la fonction Vec qui permettra d'écrire l'équation matricielle (4.1) sous la forme d'un système d'équations linéaires :

$$\begin{array}{ccc} Vec : \mathbb{C}^{m\times n} & \longrightarrow & \mathbb{C}^{mn\times 1} \\ X = \begin{pmatrix} X_{1.} \\ X_{2.} \\ \vdots \\ X_{m.} \end{pmatrix} & \rightsquigarrow & (X_{1.}, X_{2.}, \cdots, X_{m.})^T \end{array}$$

où $X_{i.}$ désigne la $i^{ème}$ ligne de la matrice X

Donc Vec fait correspondre à une matrice $X \in \mathbb{C}^{m\times n}$ un vecteur $Vec(X) \in \mathbb{C}^{mn}$.

• Rapelons la définition du produit tensoriel de deux matrices :

Soit A une matrice $m \times n$
B une matrice $p \times q$

Alors $A \otimes B$ est la matrice $mp \times nq$ définie par

$$A \otimes B = \left(\begin{array}{c|c|c|c} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ \hline a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{array}\right) \tag{4.4}$$

4.2 Equation de Lyapunov

4.2.1 Ecriture de l'équation de Lyapunov sous forme d'un système d'équations linéaires

Tout d'abord le lemme suivant nous donne une propriété de l'opérateur Vec. Ensuite la proposition 4.2.1 permet de montrer l'équivalence entre l'équation matricielle et son écriture en un système d'équations linéaires.

Lemme 4.2.1
$\forall P \in \mathbb{C}^{m \times n}$
$\forall Q \in \mathbb{C}^{n \times p}$,
nous obtenons :

$$\begin{aligned} Vec(PQ) &= (P \otimes I_p) Vec(Q) \\ &= (I_m \otimes Q^T) Vec(P) \end{aligned}$$

Proposition 4.2.1
Soit $A \in \mathbb{C}^{N \times N}$, $D \in \mathbb{C}^{N \times N}$
Alors

$$\begin{aligned} AX + XA^T &= -D^T D \\ &\Longleftrightarrow \\ Mx &= f \end{aligned} \tag{4.5}$$

avec

$$\begin{aligned} M &= A \otimes I_N + I_N \otimes A \\ x &= (\underbrace{x_{11}, \ldots, x_{N1}}_{\text{1ère colonne de x}}, \ldots, \underbrace{x_{1N}, \ldots, x_{NN}}_{\text{nème colonne de x}})^T = Vec(X) \\ f &= Vec(-D^T D) \end{aligned}$$

<u>**Preuve**</u>

Nous écrivons l'équation matricielle $AX + XA^T = -D^T D$ en son équivalent vectoriel, c'est-à-dire :

$$\begin{aligned} Vec(AX + XA^T) &= Vec(-D^T D) \\ &\Leftrightarrow \\ Vec(AX) + Vec(XA^T) &= Vec(-D^T D) \end{aligned}$$

Par le lemme 4.2.1, nous avons

$$(A \otimes I_N)Vec(X) + (I_N \otimes A)Vec(X) = Vec(-D^T D)$$

En posant $x = Vec(X)$
$f = Vec(-D^T D)$
$M = A \otimes I_N + I_N \otimes A$
nous avons bien $Mx = f$.

$\square$

Explicitons le système (4.5) et regardons plus en détail ce que représente M :

$$I_N \otimes A = \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{pmatrix}$$

$$A \otimes I_N = \left(\begin{array}{c|c|c|c} a_{11}I & a_{12}I & \cdots & a_{1N}I \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{N1}I & a_{N2}I & \cdots & a_{NN}I \end{array}\right)$$

ce qui donne :

$$M = \left(\begin{array}{cccc|c|cccc} 2a_{11} & a_{12} & \cdots & a_{1N} & \cdots & a_{1N} & 0 & \cdots & 0 \\ a_{21} & a_{11}+a_{22} & \cdots & \vdots & \cdots & 0 & a_{1N} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{11}+a_{N1} & \cdots & a_{1N} & 0 & \cdots & a_{1N} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline a_{N1} & 0 & \cdots & 0 & \cdots & a_{NN}+a_{11} & a_{12} & \cdots & 0 \\ 0 & a_{N1} & \cdots & 0 & \cdots & a_{21} & a_{NN}+a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{N1} & \cdots & a_{N1} & \cdots & \cdots & 2a_{NN} \end{array}\right)$$

En conclusion, nous avons,

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \left(\begin{array}{c|c|c} x_{11} & \cdots & x_{1N} \\ \vdots & \vdots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{array} \right) + \begin{pmatrix} \mathrm{X} & \end{pmatrix} \begin{pmatrix} A^T & \end{pmatrix} = -D^T D$$

$$\Leftrightarrow$$

$$\begin{pmatrix} \mathrm{M} \end{pmatrix} \left(\begin{array}{c} x_{11} \\ \vdots \\ x_{N1} \\ \hline \vdots \\ \vdots \\ \vdots \\ \hline x_{N1} \\ \vdots \\ x_{NN} \end{array} \right) = \begin{pmatrix} f \end{pmatrix}$$

4.2.2 Unicité de la solution de l'équation matricielle de Lyapunov

Il faut remarquer que l'équation de Lyapunov $AX + XA^T = -DD^T$ est un cas particulier de l'équation de Sylvester $AX - XB = C$ (avec $B \in \mathbb{C}^{M \times M}$ et $C \in \mathbb{C}^{N \times M}$). Le théorème qui suit est une condition nécessaire et suffisante d'unicité de la solution de l'équation de Sylvester. Le corollaire transpose ce résultat à l'équation de Lyapunov.

Théorème 4.2.1

$$\begin{array}{ccc} AX - XB = C & \text{possède} & \text{une solution unique} \\ & \Leftrightarrow & \\ \sigma(A) \cap \sigma(B) & = & \{\} \end{array}$$

<u>**Preuve**</u> :
Considérons la forme vectorielle de $AX - XB = C$,
c'est-à-dire l'équation$(A \otimes I_N - I_M \otimes B^T)x = c$.
Cette équation admet une solution unique si et seulement si la matrice $(A \otimes I_N - I_M \otimes B^T)$ est non singulière, c'est-à-dire si et seulement si ses valeurs propres sont non nulles.
Or nous savons que les valeurs propres de $(A \otimes I_N - I_M \otimes B^T)$ sont les nombres $\lambda - \mu$ avec $\lambda \in \sigma(A), \mu \in \sigma(B)$ [10].
Par conséquent, la solution de $AX - XB = C$ est unique si et seulement si $\forall \lambda \in \sigma(A), \forall \mu \in \sigma(B) : \lambda \neq \mu$.

□

Corollaire 4.2.1
L'équation de Lyapunov $AX + XA^T = -DD^T$ admet une solution unique

$$\Leftrightarrow$$

$$\sigma(A) \cap \sigma(-A) \;=\; \{\}$$

<u>**Preuve**</u> :
Nous devons vérifier le théorème 4.2.1 avec $B = -A^T$,
c'est-à-dire $\sigma(A) \cap \sigma(-A^T) = \{\}$.
Ceci est vérifié car $\sigma(A^T) = \sigma(A)$, et $\sigma(A) \cap \sigma(-A) = \{\}$.

□

Théorème 4.2.2
Si X solution de $AX + XA^T = -DD^T$
et $\sigma(A) \bigcap \sigma(-A) = \{\}$
Alors X est symétrique.

<u>**Preuve**</u> :
Prenons le symétrique de l'équation de Lyapunov :

$$\begin{aligned} (AX + XA^T) &= -DD^T \\ X^T A^T + AX^T &= -DD^T \end{aligned}$$

ce qui implique que X^T est solution.
Par unicité de la solution (voir corollaire 4.2.1), nous obtenons que $X = X^T$.

□

4.3 Opérateur de Lyapunov et son adjointe

L'adjointe de l'opérateur de Lyapunov nous sera utile pour l'application de l'algorithme QMR à l'équation matricielle de Lyapunov. En effet, dans l'algorithme de biorthogonalisation de Lanczos pour les systèmes d'équations linéaires, nous travaillons avec A^* dans les récurrences (1.10). Donc dans l'algorithme pour l'équation matricielle de Lyapunov, il faudra que nous connaissions l'adjointe de l'opérateur de Lyapunov.
Il faut trouver l'adjointe de L (défini en (4.3)), c'est-à-dire
L^* avec $L^* : \mathbb{C}^{N \times N} \longrightarrow \mathbb{C}^{N \times N}$ tel que :

$$< X, L(Y) >=< L^*(X), Y >$$

Nous travaillons donc avec le produit scalaire suivant :

$$\begin{aligned} \mathbb{C}^{N \times N} \times \mathbb{C}^{N \times N} &\longrightarrow \mathbb{C} \\ (X, Y) &\rightsquigarrow < X, Y >_1 = trace(X^T Y) \end{aligned} \tag{4.6}$$

Il faut remarquer la correspondance entre le produit scalaire euclidien (utilisé dans le cadre des algorithmes pour les systèmes d'équations linéaires) :

$$\begin{aligned} \mathbb{C}^{N^2} \times \mathbb{C}^{N^2} &\longrightarrow \mathbb{C} \\ (x, y) &\rightsquigarrow < x, y >_2 = \sum_{i=1}^{N^2} x_i y_i = y^T x \end{aligned}$$

et le produit scalaire défini en (4.6) (utilisé pour les algorithmes faisant intervenir une équation matricielle).

Proposition 4.3.1
Si $x = Vec(X)$ *et* $y = Vec(Y)$,
alors $< x, y >_2 = < X, Y >_1$

<u>**Preuve**</u> :
$< Vec(X), Vec(Y) >_2 = \sum_{i=1}^{N^2} (Vec(X))_i (Vec(Y))_i = \sum_{i,j=1}^{N} X_{ij} Y_{ij}$
$< X, Y >_1 = \sum_{i=1}^{n} (X^T Y)_{ii} = \sum_{i=1}^{n} \sum_{k=1}^{n} (X^T)_{ik} Y_{ki}$
Nous avons bien l'égalité.

□

Nous pouvons effectuer le même travail de comparaison avec l'opérateur de Lyapunov.
Si nous définissons $L(X) = AX + XA^T$ où $X \in \mathbb{C}^{N \times N}$
et $l(x) = (A \otimes I_N + I_N \otimes A)x$ où $x \in \mathbb{C}^{N^2}$,
nous avons le diagramme suivant :

$$\begin{tikzcd} \mathbb{C}^{N \times N} \arrow[r, "L"] \arrow[d, "Vec"'] & \mathbb{C}^{N \times N} \arrow[d, "Vec"] \\ \mathbb{C}^{N^2} \arrow[r, "l"] & \mathbb{C}^{N^2} \end{tikzcd}$$

Ce diagramme est commutatif car pour $X \in \mathbb{C}^{N \times N}$, $Vec(L(X)) = l(Vec(X))$.

Nous pouvons construire un schéma semblable pour l'adjointe de l'opérateur de Lyapunov :

$$\begin{tikzcd} \mathbb{C}^{N \times N} \arrow[r, "L^*"] \arrow[d, "Vec"'] & \mathbb{C}^{N \times N} \arrow[d, "Vec"] \\ \mathbb{C}^{N^2} \arrow[r, "l^*"] & \mathbb{C}^{N^2} \end{tikzcd}$$

Nous avons donc bien la même correspondance entre l'adjointe de l et l'adjointe de L. Ces schémas sont importants car ils permettent de voir la correspondance entre le "travail effectué" à partir de l'équation matricielle et celui réalisé en partant du sytème d'équations linéaires obtenu par l'opérateur Vec.

Calcul de l'adjointe

$$\begin{aligned} < X, L(Y) > &= < X, AY + YA^T > \quad \text{par définition de L} \\ &= trace(X^T(AY + YA^T)) \quad \text{par}(4.6) \\ &= trace(X^TAY + X^TYA^T) \\ &= trace(X^TAY + A^TX^TY) \end{aligned}$$

par les propriétés des traces.

Ainsi

$$\begin{aligned} < X, L(Y) > &= trace((X^TA + A^TX^T)Y) \\ &= < A^TX + XA, Y > \end{aligned}$$

ce qui montre que

$$L^* : \begin{cases} \mathbb{C}^{N\times N} & \longrightarrow \mathbb{C}^{N\times N} \\ X & \rightsquigarrow A^TX + XA \end{cases} \tag{4.7}$$

4.4 Les vecteurs de Lanczos

Nous utiliserons l'algorithme QMR décrit au chapitre 2 pour la résolution du système d'équations linéaires : $Mx = f$ avec $M = A \otimes I_N + I_N \otimes A$.
Nous prendrons directement le cas particulier où la matrice B définissant le produit scalaire (4.6) vaut I.
L'algorithme de biorthogonalisation engendre, à partir d'une paire de matrices non orthogonales entre elles, deux suites finies de matrices caractérisées par :

$$\begin{array}{ll} \bullet & V_n \in \mathbb{C}^{N\times N} \\ & V_n \in K_{n+1}(L, V_0) = span\{V_0, L(V_0), \cdots\} \\ & W_n \in \mathbb{C}^{N\times N} \\ & W_m \in K_{m+1}(L^*, W_0) = span\{W_0, L^*(W_0), \cdots\} \qquad (4.8) \\ Avec & L \text{ défini en } (4.2) \ et \ L^* \ en \ (4.7) \\ \bullet & < W_i, V_j >= 0 \ pour \ i \neq j \ et \ i, j < \nu \qquad (4.9) \end{array}$$

Rapelons la notation : $K_n(L, V_0) = K_n$
$K_n(L^*, W_0) = \tilde{K}_n$

Nous avons la même propriété qu'au chapitre 1 (propriété 1.1.1) :

Proposition 4.4.1
Les sous-espaces K_n et $\tilde{K}_n$ ainsi que les vecteurs V_n et W_n définis par (4.8) satisfont à :

1.
$$\begin{aligned} K_n &\subseteq K_{n+1}, \quad n = 1, 2, \cdots \\ \tilde{K}_n &\subseteq \tilde{K}_{n+1}, \quad n = 1, 2, \cdots \end{aligned} \tag{4.10}$$

2.
$$\begin{aligned} K_{n+1} &= span\{V_0, V_1, \cdots, V_n\}, \quad n = 0, 1, \cdots \\ \tilde{K}_{n+1} &= span\{W_0, W_1, ..., W_n\}, \quad n = 0, 1, \cdots \end{aligned} \tag{4.11}$$

3.
$$\begin{aligned} &\forall W \in \tilde{K}_n : \ < W, V_n >_B = 0, \quad 0 < n < \nu \\ &\forall V \in K_n : \ < W_n, V >_B = 0, \quad 0 < n < \nu \end{aligned} \tag{4.12}$$

4.
$$\begin{aligned} V_n &\in K_{n+1} \backslash K_n, \quad 0 < n < \nu \\ W_n &\in \tilde{K}_{n+1} \backslash \tilde{K}_n, \quad 0 < n < \nu \end{aligned} \tag{4.13}$$

La preuve est similaire à celle de la proposition 1.1.1, étant donné que nous avons le même genre d'hypothèses. Mais ici, nous travaillons avec le produit scalaire (4.6).
L'algorithme de biorthogonalisation, appliqué à la matrice $M \in \mathbb{C}^{N^2 \times N^2}$ décrite à la page 44, utilise les relations de récurrences suivantes (voir (1.10)) :

$$M v_n = v_{n+1}\gamma_n + v_n\alpha_n + v_{n-1}\beta_{n-1}$$

où les vecteurs $v_i \in \mathbb{C}^{N^2}$.

En définissant la matrice $V_i \in \mathbb{C}^{N \times N}$ par :

$$V_i = Vec^{-1}(v_i),$$

et en remarquant que pour toute matrice $X \in \mathbb{C}^{N \times N}$, on a $Vec(L(X)) = MVec(X)$, ces relations de récurrence peuvent se réécrire comme indiqué dans la proposition suivante.

Proposition 4.4.2
Les matrices V_n et W_n satisfont à une récurrence à trois termes :

$$L(V_n) = \gamma_n V_{n+1} + \alpha_n V_n + \beta_{n-1} V_{n-1} \quad n = 0, \cdots, \nu - 1 \tag{4.14}$$

avec $\alpha_n, \gamma_n, \beta_{n-1}$ définis par

$$T_n = \begin{pmatrix} \alpha_0 & \beta_0 & 0 & \cdots & 0 \\ \gamma_0 & \alpha_1 & \beta_1 & \ddots & 0 \\ 0 & \gamma_1 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \ddots & \beta_{n-2} \\ 0 & 0 & \cdots & \gamma_{n-2} & \alpha_{n-1} \end{pmatrix}$$

et nous posons V_{-1} et W_{-1} des matrices nulles et $\beta_{-1} = 0$
Nous avons une récurrence pour les W_n comme suit :

$$L^*(W_n) = \tilde{\gamma}_n W_{n+1} + \tilde{\alpha}_n W_n + \tilde{\beta}_{n-1} W_{n-1} \quad j = 0, \cdots, \nu - 1 \tag{4.15}$$

avec $\tilde{\alpha}_n, \tilde{\gamma}_n, \tilde{\beta}_{n-1}$ définis par $\tilde{T}_n$ qui est la matrice T_n avec des tildes sur chaque élément.

Corollaire 4.4.1
Nous pouvons réécrire (4.14) de la façon suivante :

$$\underbrace{[L(V_0)| \cdots |L(V_{n-1})]}_{(N,nN)} = \underbrace{[V_0 \cdots V_n]}_{(N,(n+1)N)} \underbrace{(\underline{T_n} \otimes I_N)}_{((n+1)N,nN)} \tag{4.16}$$

avec

$$\underline{T_n} = \left(\begin{array}{ccccc} \alpha_0 & \beta_0 & 0 & \cdots & 0 \\ \gamma_0 & \alpha_1 & \beta_1 & \ddots & 0 \\ 0 & \gamma_1 & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & \gamma_{n-2} & \alpha_{n-1} \\ \hline 0 & \cdots & \cdots & 0 & \gamma_{n-1} \end{array}\right) \in \mathbb{C}^{(n+1)\times n}$$

4.5 L'algorithme de biorthogonalisation de Lanczos

Les relations de récurrence données par la proposition 4.4.2 permettent d'écrire l'algorithme de biorthogonalisation appliqué directement à l'écriture matricielle.

– <u>Initialisation</u>

$$
\begin{array}{ll}
\text{choisir } V_0, W_0 \text{ tels que } \delta_0 =< W_0, V_0 >= tr(W_0^T V_0) \neq 0 & (1) \\
\qquad \beta_{-1} = 0 & (2) \\
n = 0 &
\end{array}
$$

– <u>Itération $n+1$</u>

$$
\begin{array}{ll}
\alpha_n = \frac{<W_n, L(V_n)>_B}{\delta_n} \quad ; \tilde{\alpha_n} = \overline{\alpha_n}, & (3) \\
\text{IF } n > 0 \text{ THEN} & \\
\qquad \beta_{n-1} = \overline{\tilde{\gamma}_{n-1}}\delta_n/\delta_{n-1} & \\
\qquad \tilde{\beta}_{n-1} = \overline{\gamma_{n-1}\delta_n/\delta_{n-1}} = \overline{\beta_{n-1}\gamma_{n-1}/\tilde{\gamma}_{n-1}}, & (4) \\
\text{ENDIF} & \\
V_{temp} = L(V_n) - V_n\alpha_n - V_{n-1}\beta_{n-1} & \\
W_{temp} = L^*(W_n) - W_n\tilde{\alpha}_n - W_{n-1}\tilde{\beta}_{n-1} & (5) \\
\delta_{temp} =< W_{temp}, V_{temp} > & (6)
\end{array}
$$

– <u>Test</u>

$$
\begin{array}{ll}
\text{IF } \delta_{temp} = 0 \quad THEN & (7) \\
\quad \text{choisir } \gamma_n \neq 0, \tilde{\gamma}_n \neq 0 & \\
\quad \nu = n + 1 & \\
\quad V_\nu = \frac{V_{temp}}{\gamma_n} & \\
\quad W_\nu = \frac{W_{temp}}{\tilde{\gamma}_n} & (8) \\
\quad \delta_{n+1} = 0 & (9) \\
\quad \text{STOP} & \\
\text{ELSE} & \\
\quad \text{choisir} \quad \gamma_n \neq 0, \tilde{\gamma}_n \neq 0 & \\
\qquad\qquad \delta_{n+1} \text{ tel que } \gamma_n\overline{\tilde{\gamma}_n}\delta_{n+1} = \delta_{temp} & \\
\quad V_{n+1} = \frac{V_{temp}}{\gamma_n} & (10) \\
\quad W_{n+1} = \frac{W_{temp}}{\tilde{\gamma}_n} & (11) \\
\quad \text{n=n+1} &
\end{array}
$$

4.6 L'algorithme QMR

Le problème à résoudre est le suivant :

$$\begin{aligned} AX + XA^T &= -DD^T \ voir(4.1) \\ &\Longleftrightarrow \\ L(X) &= F \ voir(4.3) \\ &\Longleftrightarrow \\ Mx &= f \ voir(4.5) \end{aligned}$$

Le résidu s'écrit :

$$R_n = F - L(X_n)$$

Le $n^{ème}$ itéré X_n est choisi tel que :

$$X_n = X_0 + K_n(\underbrace{R_0}_{N\times N}, L) \Longleftrightarrow$$

$$X_n = X_0 + \underbrace{[\underbrace{V_0}_{N\times N} \cdots V_{n-1}]}_{N\times Nn}\underbrace{(\underbrace{k_n}_{n\times 1} \otimes I_N)}_{Nn\times N} \tag{4.17}$$

avec $k_n \in \mathbb{C}^n$
et $V_0, \cdots, V_{n-1}$ obtenus par l'algorithme de biorthogonalisation de Lanczos.

Nous initialisons V_0 comme suit :

$$V_0 = \frac{R_0}{||R_0||_F} \tag{4.18}$$

Nous voulons minimiser la norme du résidu.
Regardons comment écrire le résidu pour l'équation matricielle (4.3).

Proposition 4.6.1
Nous pouvons écrire le résidu comme suit :

$$R_n = [V_0 \cdots V_n][(e_1 \otimes I_N)||R_0||_F - (\underline{T_n} k_n \otimes I_N)] \tag{4.19}$$

Preuve :
Nous avons :

$$R_n = F - L(X_n)$$

En utilisant (4.17), nous obtenons :

$$L(X_n) = L(X_0) + L([V_0 \cdots V_{n-1}](k_n \otimes I_N))$$

et

$$R_n = F - L(X_0) - L([V_0 \cdots V_{n-1}](k_n \otimes I_N))$$

avec $[V_0 \cdots V_{n-1}](k_n \otimes I_N) = k_{n,0}V_0 + k_{n,1}V_1 + \cdots + k_{n,(n-1)}V_{n-1}$

en prenant $k_n = \begin{pmatrix} k_{n,0} \\ \vdots \\ k_{n,(n-1)} \end{pmatrix}$

Nous pouvons réécrire le résidu :

$$R_n = R_0 - k_{n,0}L(V_0) - k_{n,1}L(V_1) - \cdots - k_{n,(n-1)}L(V_{n-1})$$

En utilisant la récurrence à trois termes (4.14) :

$$R_n = R_0 - [V_0 \cdots V_n](\underline{T_n} \otimes I_N)(k_n \otimes I_N)$$

Par les propriétés du produit tensoriel : $(\underbrace{\underline{T_n} \otimes I_N}_{((n+1)N,nN)})\underbrace{(k_n \otimes I_N)}_{(nN,n)} = \underbrace{(\underline{T_n}k_n \otimes I_N)}_{((n+1)N,n)}$

Nous réécrivons R_n :

$$R_n = R_0 - [V_0 \cdots V_n](\underline{T_n}k_n \otimes I_N) \tag{4.20}$$

De plus, par le choix de la matrice initiale (4.18), nous avons :

$$\begin{aligned} R_0 &= [V_0|\cdots|V_n]\left[\begin{array}{c} \begin{matrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{matrix} \\ \hline 0 \\ \hline 0 \end{array}\right]||R_0||_F \\ &= [V_0|\cdots|V_n](e_1 \otimes I_N)||R_0||_F \end{aligned}$$

En remplaçant R_0 par cette expression dans (4.20), nous obtenons exactement (4.19) .

$\square$

Comme au chapitre 2, le problème $\min_{k_n \in \mathbb{C}^n} ||R_n||_F$ est remplacé par :

$$\min_{k_n \in \mathbb{C}^n} || \; \| R_0 \|_F \; e_1 \otimes I_N - \underline{T_n} k_n \otimes I_N ||_2 \tag{4.21}$$

Or $\| R_0 \|_F \; e_1 \otimes I_N - \underline{T_n} k_n \otimes I_N = (\| R_0 \|_F \; e_1 - \underline{T_n} k_n) \otimes I_N$ Et
$|| \; \| R_0 \|_F \; e_1 \otimes I_N - \underline{T_n} k_n \otimes I_N ||_2 = || \; \| R_0 \|_F \; e_1 - \underline{T_n} k_n ||_2$ ([10]),
ainsi il est équivalent de traiter le problème suivant :

$$\min_{k_n \in \mathbb{C}^n} || \; \| R_0 \|_F \; e_1 - \underline{T_n} k_n ||_2 \tag{4.22}$$

Nous pouvons donner le principe général comme suit :

étape 0

choix de X_0
calcul de $R_0 = F - L(X_0)$

étape 1

a) Effectuer la première étape de l'algorithme de biorthogonalisation :

- choix de $V_0 = \dfrac{R_0}{||R_0||_F}$ et de W_0 tq $||W_0|| = 1$
- détermination de $\underline{T_1} \in \mathbb{C}^2$

b) Déterminer $X_1 = X_0 + V_0(k_1 \otimes I_N)$ en résolvant $\min_{k_1 \in \mathbb{C}} || \; \| R_0 \|_F - \underline{T_1} k_1 ||^2$
c) Test d'arrêt :

- calcul de R_1
- Si X_1 satisfait à la condition d'arrêt, nous nous arrêtons, sinon, nous passons à l'étape 2, et ainsi de suite...

4.7 Itérés symétriques

Par le théorème 4.2.2, nous avons que si $AX + XA^T = -DD^T$ admet une solution unique, alors elle est symétrique. Il est donc intéressant de regarder si nos itérés sont symétriques quand ils sont générés par l'algorithme QMR.
Tout d'abord, nous allons faire la preuve que QMR, appliqué à l'équation $H(X) = F$ où H est un opérateur linéaire, génère des itérés symétriques. Ensuite, nous vérifierons si le théorème est valable quand nous appliquons QMR pour résoudre l'équation de Lyapunov.

Théorème 4.7.1
Soit un opérateur $H : \mathbb{C}^{N\times N} \longrightarrow \mathbb{C}^{N\times N}$ linéaire
Soit X_0, F sont symétriques et $\in \mathbb{C}^{N\times N}$
Si pour une matrice symétrique $X \in \mathbb{C}^{N\times N}$, nous avons :

$$\begin{aligned} H(X) &= [H(X)]^T \\ H^*(X) &= [H^*(X)]^T \end{aligned} \tag{4.23}$$

alors QMR , appliqué à $H(X) = F$, génère des itérés X_n symétriques.

<u>**Preuve**</u> :

1. <u>V_0 est symétrique</u>

   - R_0 est symétrique
     car, par définition, $R_0 = F - H(X_0)$
     Et nous devons montrer que

     $$(F - H(X_0))^T \quad = \quad (F - H(X_0))$$

     c'est-à-dire $F^T - [H(X_0)]^T = F - H(X_0)$
     Ce qui est vérifié par (4.23) et le fait que F, X_0 soient symétriques par hypothèse.
   - V_0 est symétrique vu que $V_0 \in span\{R_0\}$

2. <u>X_n est symétrique</u>

   Nous avons que $X_n \in X_0 + K_n(R_0, H)$, ce qui peut s'écrire :
   $X_n = X_0 + \alpha_1 V_0 + \alpha_2 H(V_0) + \cdots + \alpha_{n-1} H^{n-1}(V_0)$ et
   $X_n^T = X_0^T + \alpha_1 V_0^T + \alpha_2 [H(V_0)]^T + \cdots + \alpha_{n-1}(H^{n-1}(V_0))^T$.
   Par hypothèse, $X_0 = X_0^T$ est vérifié.

Le point 1. de la preuve nous montre que $V_0 = V_0^T$. En utilisant (4.23) et le point 1., nous obtenons que $H(V_0) = [H(V_0)]^T$.
Pour finir, en sachant que $H^i = H(H^{i-1})$, par récurrence, nous obtenons que $H^i(V_0) = (H^i(V_0))^T$ pour $i = 2, \cdots, n-1$. Ainsi, nous avons bien la symétrie des itérés X_n.

□

Corollaire 4.7.1
Si X_0 sont symétriques,
alors la méthode QMR appliquée à l'équation de Lyapunov génère des itérés X_n symétriques.

Preuve :
Nous avons en transposant $L(X) = AX + XA^T$:

$$\begin{aligned} L^T(X) &= (AX + XA^T)^T \\ &= X^T A^T + AX^T \\ &= XA^T + AX \quad \text{vu que X symétrique} \\ &= L(X) \end{aligned}$$

Nous pouvons donc bien appliquer le théorème 4.7.1 à l'équation de Lyapunov.

□

4.8 Préconditionnement

Dans [12], M. Hachbuck et G. Starke ont étudié la résolution de l'équation de Lyapunov par la méthode QMR. Ils ont, entre autres, montré qu'il est essentiel d'utiliser une technique de préconditionnement en association avec la méthode QMR. Les résultats numériques qu'ils obtiennent sur deux exemples précis avec les techniques de préconditionnement ADI et SSOR montrent que pour une précision fixée, le nombre d'itérations est nettement moindre lorsqu'on utilise un préconditionnement. Le choix de ces deux préconditionnements est justifié par le fait que ces techniques, associées à la méthode QMR, conservent aux itérés leur caractère symétrique.

Nous n'avons pas analysé ces techniques de préconditionnement dans ce travail. Nous rappelons néanmoins quelques généralités sur le préconditionnement associé à la méthode QMR.

Le but du préconditionnement est de transformer une équation, ici nous considérons l'équation de Lyapunov $AX + XA^T = -DD^T$, en une équation plus

facile à résoudre. Cela peut se faire de la façon suivante :
nous multiplions l'équation de Lyapunov par la gauche par une matrice non singulière $Q_1 \in \mathbb{C}^{N\times N}$, et par la droite par une matrice non singulière $Q_2 \in \mathbb{C}^{N\times N}$.
L'équation de Lyapunov devient

$$\mathbf{Q_1} A Q_1^{-1} Q_1 X \mathbf{Q_2} + \mathbf{Q_1} X Q_2 Q_2^{-1} A^T \mathbf{Q_2} = -\mathbf{Q_1} D^T D \mathbf{Q_2} \tag{4.24}$$

Nous obtenons ainsi une équation de Sylvester de la forme :

$$\tilde{A}\tilde{X} + \tilde{X}\tilde{B} = -Q_1 D^T D Q_2 \tag{4.25}$$

avec $\tilde{A} = Q_1 A Q_1^{-1}$
$\tilde{B} = Q_2^{-1} A^T Q_2$
$\tilde{X} = Q_1 X Q_2$

Nous pouvons prendre le cas particulier où $Q_2 = Q_1^T$ pour obtenir une équation de Lyapunov :

$$\tilde{A}\tilde{X} + \tilde{X}\tilde{A}^T = -\tilde{D}^T\tilde{D} \tag{4.26}$$

avec $\tilde{A} = Q_1 A Q_1^{-1}$
$\tilde{X} = Q_1 X Q_1^T$
$\tilde{D} = D Q_1^T$

Pour la suite, nous choisirons le préconditionnement où $Q_2 = Q_1^T$.
Il est intéressant de voir le lien entre l'équation de Lyapunov initiale $AX+XA^T = -DD^T$ et l'équation de Lyapunov préconditionnée (4.26) .

Soit $K_n(L, R_0) = span\{R_0, L(R_0), \cdots, L^{n-1}(R_0)\}$
$K_n(\tilde{L}, \tilde{R}_0) = span\{\tilde{R}_0, \tilde{L}(\tilde{R}_0), \cdots, \tilde{L}^{n-1}(\tilde{R}_0)\}$
avec $R_0 = -DD^T - L(X_0)$ et $\tilde{R}_0 = -\tilde{D}^T\tilde{D} - \tilde{L}(\tilde{X}_0)$

et $L(X) = AX + XA^T$
$\tilde{L}(\tilde{X}) = \tilde{A}\tilde{X} + \tilde{X}\tilde{A}^T$

Vu que nous voulons minimiser la norme du résidu, regardons le lien entre les résidus correspondant aux équations (4.1) et (4.26).

Proposition 4.8.1
Nous avons les deux égalités suivantes :

$$\tilde{R}_0 = Q_1 R_0 Q_1^T \tag{4.27}$$
$$\tilde{R}_n = Q_1 R_n Q_1^T \tag{4.28}$$

Preuve :
Par la définition de $\tilde{L}$, nous pouvons écrire :

$$\begin{aligned}\tilde{R}_0 &= -Q_1 D^T D Q_1^T - \tilde{A}\tilde{X}_0 - \tilde{X}_0\tilde{A}^T \\ &= -Q_1 D^T D Q_1^T - (Q_1 A Q_1^{-1})(Q_1 X_0 Q_1^T) - (Q_1 X_0 Q_1^T)(Q_1 A^{-1} Q_1^T) \\ &= Q_1[-DD^T - AX_0 - X_0 A^T]Q_1^T \\ &= Q_1 R_0 Q_1^T\end{aligned}$$

□

Nous allons utiliser cette dernière proposition pour analyser le lien entre les sous-espaces de Krylov.

Proposition 4.8.2
Les sous-espaces de Krylov $K_n(L, R_0)$ et $\tilde{K}_n(\tilde{L}, \tilde{R}_0)$ satisfont à l'égalité suivante :

$$K_n(L, R_0) = Q_1^{-1}\tilde{K}_n(\tilde{L}, \tilde{R}_0)Q_1^{-T} \tag{4.29}$$

Preuve :
Par la définition des sous-espaces de Krylov, il est suffisant de montrer que :

$$Q_1(L^k(R_0))Q_1^T = \tilde{L}^k(\tilde{R}_0) \quad pour \quad k = 1, \cdots, n-1 \tag{4.30}$$

La preuve se fait par induction.

•Pour $k = 1$, nous avons bien

$$Q_1(L(R_0))Q_1^T = \tilde{L}(\tilde{R}_0)$$

car

$$\begin{aligned}Q_1(L(R_0))Q_1^T &= Q_1(AR_0 + R_0A^T)Q_1^T \\ &= Q_1 A Q_1^{-1} R_0 Q_1^T + Q_1 R_0 Q_1^T Q_1^{-1} A^T Q_1^T \\ &= \tilde{A}\tilde{R}_0 + \tilde{R}_0\tilde{A}^T \text{par la proposition 4.8.1}\end{aligned}$$

• L'hypothèse est que l'égalité (4.30) est vérifiée pour k, montrons qu'elle est vérifiée pour $k+1$.

$$\begin{aligned}
Q_1(L^{k+1}(R_0))Q_1^T &= Q_1[A(L^k(R_0)) + (L^k(R_0))A^T]Q_1^T \\
&= Q_1[AQ_1^{-1}(\tilde{L}^k(\tilde{R}_0))Q_1^{-T} + Q_1^{-1}(\tilde{L}^k(\tilde{R}_0))Q_1^{-T}A^T]Q_1^T \\
&= \tilde{A}(\tilde{L}^k(\tilde{R}_0)) + (\tilde{L}^k(\tilde{R}_0))\tilde{A}^T \text{par l'hypothèse de récurrence} \\
&= \tilde{L}^{k+1}(\tilde{R}_0)
\end{aligned}$$

□

Les itérés X_n, générés par l'algorithme QMR appliqué à l'équation de Lyapunov $L(X) = F$, appartiennent à $X_0 + K_n(L, R_0)$,
et les itérés $\tilde{X}_n$, générés par l'algorithme QMR appliqué à l'équation de Lyapunov préconditionnée (4.26), appartiennent à $\tilde{X}_0 + \tilde{K}_n(\tilde{L}, \tilde{R}_0)$.
Or la proposition 4.8.2 montre qu'à chaque itété $\tilde{X}_n$ correspond un itéré $X_n = Q_1^{-1}\tilde{X}_nQ_1^{-T}$ qui appartient à $X_0 + K_n(L, R_0)$.
La propriété 4.8.1 montre que

$$Min||\tilde{R}_n||_F \Longleftrightarrow Min||Q_1R_nQ_1^T||_F \tag{4.31}$$

Si nous définissons $||X||_* = ||Q_1XQ_1^*||_F$, alors

$$Min||\tilde{R}_n||_F \Longleftrightarrow Min||R_n||_* \tag{4.32}$$

Nous minimisons donc le même résidu mais dans deux normes différentes.

Conclusion

L'algorithme QMR pour la résolution des systèmes d'équations linéaires comprend principalement deux parties :
* Générer les vecteurs de Lanczos
* Résoudre un problème au sens des moindres carrés.

Dans ce mémoire, nous avons étudié cet algorithme
* pour résoudre un système d'équations linéaires $Ax = b$
* pour résoudre l'équation matricielle de Lyapunov $AX + XA^T = -D^T D$

Pour l'algorithme QMR appliqué à un système d'équations linéaires, nous avons analysé deux implémentations. La première, décrite au chapitre 2, utilise l'algorithme de biorthogonalisation de Lanczos (chapitre 1), et résout le problème au sens des moindres carrés par une factorisation QR. La seconde est une variante qui utilise l'algorithme de biorthogonalisation basé sur un couple de récurrences à deux termes, ce qui permet de résoudre le problème de minimisation grâce à une récurrence simple et d'éviter ainsi la factorisation QR.
Il aurait été intéressant d'implémenter ces deux versions de l'algorithme QMR pour les comparer, mais le manque de temps ne nous l'a pas permis.

En ce qui concerne l'algorithme QMR appliqué à l'équation de Lyapunov, une amélioration peut être apportée en préconditionnant. La vérification des résultats obtenus par M. Hochbruck et G. Starke [9] pourrait faire l'objet d'un mémoire futur ...

Bibliographie

[1] J.-C. DESSY, *Résolution de l'équation de Sylvester par des méthodes basées sur les sous-espaces de Krylov*, mémoire aux FUNDP, Namur, 1998.

[2] C. DUFOUR, *L'algorithme de biorthogonalisation de Lanczos et ses "breakdowns"* , mémoire aux FUNDP, Namur, 1996.

[3] R.W. FREUND, G.H. GOLUB and N.M. NACHTIGAL, *Iterative solution of linear systems*, Acta Numerica, 1991, pp 60-79.

[4] R.W. FREUND, N.M. NACHTIGAL, *QMR : a quasi-minimal residual method for non-hermitian linear systems*, Numerical Math., 60, 1991, pp 315-339.

[5] R.W. FREUND, N.M. NACHTIGAL, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. SCI. COMPUT, 15, n2, 1994, pp313-337.

[6] G.H. GOLUB and C.VAN LOAN, *Matrix computations*, John Hopkins University Press,Baltimore, 1993.

[7] M.H. GUTNECHT, *Lanczos type solvers for nonsymmetric linear systems of equations*, Acta Numerica, Cambridge University Press,1997, pp 271-397.

[8] P.R. HALMOS, *Finite-Dimensional Vector Spaces*, Springer-Verlag, New-York, 1974.

[9] M. HOCHBRUCK and G. STARKE, *Preconditioned Krylov subspace methods for Lyapunov matrix equations*, SIAM J. Matrix anal. appl., 16, 1995, pp 156-171.

[10] A. HORN AND C.R. JOHNSON, *Matrix Analysis*, Cambridge Unversity Press, 1985.

[11] F. RINGLET, *La méthode ADI et son application à l'équation de Sylvester*, mémoire aux FUNDP, Namur, 1993.

[12] M. SAUREN AND H.M. BUCKER, *On deriving the quasi minimal residual method*, Siam revieuw, Vol. 40, No.4, december 1998, pp. 922-926.

ERRATA

- A la page 12, l'équation (1.28) s'écrit comme suit : $\delta_{n+1}\gamma_n = \delta_n\overline{\tilde{\beta}_n}$
- A la page 23, $c_n \in I\!R$
- A la page 24, il faut conjuguer et transposer la matrice $\begin{pmatrix} c_n & -s_n \\ \overline{s}_n & c_n \end{pmatrix}$
- A la page 26, $s_n \in \mathbb{C}$
- A la page 35, dans le lemme 3.2.1, $y_2 = (\kappa_0 + \theta_1\kappa_0, \kappa_1)^T$
- A la page 36, y_1 doit vérifier l'équation normale :

$$\begin{pmatrix} \overline{\tau_0} & \overline{\rho_1} \end{pmatrix}\begin{pmatrix} \tau_0 \\ \rho_1 \end{pmatrix} y_1 = \begin{pmatrix} \overline{\tau_0} & \overline{\rho_1} \end{pmatrix}\begin{pmatrix} \overline{\rho_0} \\ 0 \end{pmatrix}$$

- A la page 37, dans l'équation (3.23), il faut lire κ_{n-1} au lieu de κ_{i-1}
- A la page 44, le premier bloc $a_{11}I$ de la matrice M est :

$$\begin{matrix} 2a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{11}+a_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{11}+a_{NN} \end{matrix}$$

- A la page 45,

$$\left(\text{ M } \right) \left(\begin{array}{c} x_{11} \\ \vdots \\ x_{N1} \\ \hline \vdots \\ \vdots \\ \vdots \\ \hline x_{1\mathbf{N}} \\ \vdots \\ x_{NN} \end{array} \right) = \left(\; f \; \right)$$

- A la page 51, la relation (4.15) est vérifiée pour $\mathbf{n} = 0, \cdots, \nu - 1$
- A la page 53, il faut lire $X_n = X_0 + K_n(\mathbf{L}, \mathbf{R}_0)$
- A la page 57, il faut lire "X_0 est symétrique" au lieu de "X_0 sont symétriques".
- A la page 60, la définition de $||X||_*$ est $||X||_* = ||Q_1 X Q_1^{\mathbf{T}}||_F$